Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Perturbed-Analytic Direct transcription for Optimal Control (PADOC)

Amin Jafarimoghaddam* | Manuel Soler

[1]Department of Bioengineering and Aerospace Engineering, Universidad Carlos III de Madrid, Avenida de la Universidad, 30, Leganes, 28911 Madrid, Spain

Correspondence
*Amin Jafarimoghaddam, Email: ajafarim@pa.uc3m.es, Tel.: +34 666 251 627

Summary

The herein coined PADOC (Perturbed-Analytic Direct transcription for Optimal Control) stands for a new transcription method for direct trajectory optimization. It is launched on an analytic nonlinear solver that generates series solutions based on the perturbation principle. The perturbation is computed around a known solution that is linked to an arbitrary linear operator. We construct PADOC on a novel segmented decomposition method. PADOC comes along with a new solution approach, namely, average NonLinear Programming (aNLP). The aNLP theorem suggests generating staircase optimal solution (low resolution) and turning it into a distributed optimal solution (high resolution) by exploiting the Hamiltonian. The analytic architecture promotes PADOC to institute a certain analytic connection between the discrete nodes. This renders stability, accuracy within an analytic resolution (for the mathematical objects), robustness and a cut-down in the number of the decision variables in the frame of NLP. We also give a rigorous proof that the multipliers associated with Karush–Kuhn–Tucker optimality conditions in the frame of NLP, as transcribed by PADOC, correspond to a backward analytic solution for the costate equations. We finally let PADOC speak through some examples.

KEYWORDS:
Direct Optimization, Transcription Methods, Trajectory Optimization, Analytic Transcription Method, Optimal Control, Aircraft Trajectory Optimization

1 | INTRODUCTION

The XXI century has witnessed innumerable applications of optimal control theory in disparate areas of research such as biomedicine, e.g. [1], aircrafts/spacecrafts, e.g. [2],[3],[4],[5] and robotics, e.g., [6]. In this respect, the associated industries are in demand of solution methodologies for Optimal Control Problems (OCP) with high accuracy, robustness and reasonable computational time. In a conventional point of view, the methods for OCPs can be put into two categories, indirect and direct approaches [7]. The indirect approach is linked to Pontryagin Maximum Principle (PMP) leading to a system of Multi-Point Boundary Value Problem (MPBVP). In a direct approach, the transcribed/parametric version of the OCP as a NonLinear Programming (NLP) is fed to an optimization solver such as interior point or Successive Quadrative Programming (SQP). Indeed, the method of transcription affects the sparsity of the Karush–Kuhn–Tucker (KKT) system and accordingly the search direction matrix, accuracy and the required iterations in the frame of NLP. However, as long as the transcription method meets a sufficient level of accuracy

over the search space, the general structure of the NLP problem is preserved, that is, a convex OCP remains convex on dissimilar accurate transcription methods. In this respect, a rigorous mathematical analysis is still lacking to evaluate the possible repercussions on the NLP due to different transcription methods (a more detailed review on this matter can be tracked in[8]).

The advantage of the indirect approach is to provide accurate solutions. However, theoretical analysis reveals a mathematical connection/equivalence between the indirect and direct methods[9]. Generally speaking, Covector Mapping Principle (CMP) asserts that there exists a linkage between an indirect and a direct method through Hahn-Banach theorem, see e.g.,[10],[11],[12] and[13]. In other words, CMP (which is an order-preserving map between the primal and dual variables) pairs a direct method with an equivalent indirect method. This means, the solution by a direct method converges to the solution by its indirect counterpart. This understanding has led to a newly coined term, Hamiltonian programming (for more information, interested readers are referred to[14] and the relevant references therein). Moreover, CMP is regarded as a criterion to evaluate whether a transcription method is complete[9].

The present research aims to propose a novel analytic transcription approach for optimal control problems that is complete (accounting CMP criterion). The corestone of the presented method, Perturbed Analytic Direct transcription for Optimal Control (PADOC), is to provide a rigorous analytic connection between the intermediate nodes. In this sense, solution for the state variables is exact. The analytic connection can be explained by assuming two adjacent nodes from the state dynamics with one decision variable being allowed to vary in parameter optimization. In this situation, PADOC assures that the two nodes are always analytically connected over the entire search space. Therefore, PADOC is guaranteed for the convergence of the system dynamics with only a few segments for a wide class of nonlinearity (the property lacking in most of the numerical integration methods). Comparing to the collocation methods, the immediate understanding is to provide a transformed cost functional within an analytic resolution. In the sequel, we provide a state of the art critique on transcription approaches to highlight the existing gaps.

A direct trajectory optimization requires a transcription method to transform the original problem into an NLP. The transcription methods can be viewed as shooting and collocation. A circumstantial review of these methods can be found in several resources (see e.g.,[15],[16]) and this suffices to avoid superfluous description. Specifically, shooting techniques employ either explicit (such as Euler and Runge- Kutta) or implicit (such as Hermite–Simpson) approaches to discretize the system dynamics normally with piecewise constant control (see e.g.,[17] and[18]). In collocation techniques (see e.g.,[19],[20] and[21]), the system dynamics and control are considered as piecewise functions, approximated by orthogonal polynomials, and the system dynamics are satisfied at some nodes, known as the collocation points. This leads to an NLP with states being explicitly dependent on the control resulting in more memory storage[8]. As long as the stability of the solution to the system dynamics is concerned, collocation methods are more advantageous over the shooting methods. This statement is supported by referring to the fact that in collocation methods, the dynamics' path between the imposed handling nodes are governed/interpolated by some predefined base functions of polynomial types. This is while shooting methods need to estimate the dynamics' path by recursive numerical drifts. Therefore, stability of the shooting methods is restricted to the stability of the numerical discretization. On the other hand, solutions by the collocation methods are valid only at the collocation points. A drawback of the shooting methods is the need for a prior knowledge on the sufficient discrete segments to guarantee the convergence of the system dynamics. Therefore, the number of grids is normally considered large enough to guarantee convergence, leading to extra memory storage. Besides, the sole concept of convergence, apart from the accuracy, can be loosely interpreted in such a way that the convexity of the OCP is preserved. However, the Hessian search direction matrix is altered affecting the required number of Newton-type iterations to find the optimal solution with a limited accuracy. One of the underlying reasons associated with the failure of the optimization module is that a numerical discretization cannot guarantee that the system dynamics are satisfied in iterative steps performed by the optimization module. In the case of singular OCPs, the Hessian matrices are ill-posed. Experience shows that this ill-conditioning is magnified as the number of nodes increases. This is whilst, there are many practical nonlinear OCPs requiring a large number of nodes in order to secure the convergence of the system dynamics mostly in the frame of the numerical shooting transcriptions. An amalgamation of these statements indicates a plausible failure of the shooting methods in these circumstances. Therefore, both collocation and shooting methods present well-identified drawbacks that we claim to (at least partially) overcome with PADOC for different classes of OCPs.

A more challenging situation is the presence of singular arcs in the OCP, leading to the ill-posedness of the Hessian matrix. In other words, the projected Hessian matrix is not positive definite[22]. This ill-conditioning shows itself as an oscillatory or fluctuating response in the singular region. There are some reported strategies to overcome this situation, e.g., mesh refinement and hubristic strategies (see e.g.,[23],[24],[25],[26],[27]), combining indirect and direct pathways (see e.g.,[28],[29],[30]), parameter continuation[31]

(sometimes referred as homotopy) or a combination of these strategies [32]. More specifically, [25] suggests a mesh refinement strategy to increase the resolution of a singular arc solution in a direct method. The nested direct transcription optimization as in [26], splits the OCP into an inner and an outer OCP and finds an optimized mesh distribution. In [30], the authors proposed a strategy that amalgamates indirect and direct methods. More specifically, the OCP is initially solved by direct optimization to extract the solution structure and to estimate the onsets of the costate variables. When it comes to parameter continuation [31], the strategy is to switch a singular OCP to a nonsingular OCP. Basically, parameter continuation consists of smooth deformations from a known (or easy to solve) OCP to the problem of interest in infinitesimal steps such that the solution of each step serves as a meaningful initial guess for the next step. In this way, parameter continuation is recommended as a refinement strategy in a direct or an indirect method. [32], contains mesh refinement processing as well as a regularization procedure with an embedding parameter to avoid ill-conditioning of the Hessian matrix by virtually turning a singular problem into a nonsingular one.

In a recent publication [33], an analytic transcription method presumably as the first appearing attempt in the literature for trajectory optimization problems has been presented. The method has been launched upon Homotopy Analysis Method (HAM) that is a powerful analytic nonlinear solver over about two decades (interested readers are referred to some original references such as (see e.g., [34], [35], [36], [37])). However, it certainly cannot guarantee a global convergence for the nonlinearities. It is due to the fact that HAM normally seeks a unified solution for the entire domain of interest. Therefore, the convergence of HAM is reliant on the existence of a fixed point for the inverse of the linear operator as it applies to the nonlinearity over the entire domain. Since the choice for the linear operator is free in HAM and there is no theory to restrict this choice, it does not guarantee a global convergence. In this respect, authors have treated different nonlinearities with different linear operators through HAM on a case by case basis (see e.g., [34], [36] and [38]). Therefore, the method proposed in [33] will fail as it confronts strongly-nonlinear system dynamics. As a consequence, PADOC stands as the first robust analytic transcription method in the literature.

We resort now our discussion to the assessment of the completeness of PADOC using the CMP principle [9]. Connection between the multipliers associated with Karush–Kuhn–Tucker (KKT) optimality conditions and the continuous costate functions has been investigated in the past in several resources such as [39], [40] and [41]. For some transcription methods, such as the ones in [39] and [40], the multipliers are connected to a discrete numerical solution for the costate variables (dual variables) with a reduced order of accuracy compared to the accuracy of the state variables (primal variables). For the Legendre–Gauss–Lobatto (LGL) collocation, as in [41], the costates at the LGL points are equal to the KKT multipliers divided by the LGL weights. Here, we prove that the KKT multipliers at the SADM nodes correspond to a backward analytic solution of the costate equations. Moreover, we prove that the cost functional in the frame of NLP is decomposed in a pure analytic manner.

Therefore, the novel contributions of the present paper can be viewed as: 1) presenting SADM as a new analytic nonlinear solver and 2) a new NLP solving strategy, namely average NonLinear Programming (aNLP). SADM and aNLP are exploited to form a novel analytic transcription method, PADOC.

PADOC has the following attributes: a) to be hp adaptive. b) to have the minimum number of nodes. c) to show marginal sensitivity to the initial guess due to the analytic connection between the imposed nodes. d) to show highly accurate estimations for the three main mathematical objects, namely, the system dynamics, constraints and the cost functional. e) to be capable of handling singular OCPs. f) to exhibit a quite spare programming for regular OCPs (no bang/singular structures). g) to have primal-dual consistency (CMP completeness).

Finally, it is worth mentioning that, in general, PADOC could be launched upon other analytic nonlinear solvers such as HAM. However, for the sake of convergence, it is a mandate to design a segmented form of HAM. Even though the usage of HAM within PADOC lays outside the present paper. We will aim in the future to present different forms of PADOC with possible distinguished merits.

The paper is constructed as follows: In Sec.2 we present ADM, as a prominent analytic nonlinear solver. In Sec.3, we design SADM as a new analytic solver together with the supportive theorems 1 and 2 to assert the unconditionally-convergent argument. Sec.4 is devoted to the mathematical representation of SADM in OCPs. In Sec.5 we present the primal-dual consistency of PADOC together with the aNLP theorem. In Sec. 6 the PADOC algorithm is given. In Sec.7 we solve some examples with varying complexity to show the applicability and robustness of PADOC. These examples are: 1) an orbital transfer problem (7.1); 2) the Van der Pol oscillator (7.2); 3) Brachistochrone problem (7.3); 4) Aly problem (7.4); 5) maximum range of a hang glider (7.5); 6) minimum time aircraft trajectory in climbing phase (7.6).

2 | ADOMIAN DECOMPOSITION METHOD

Let us consider a nonlinear problem in a general form as ,

$$\mathcal{L}\big[x(t)\big] = \mathcal{N}\big[x(t)\big] + \mathcal{R}\big[x(t)\big] + g(t) \quad t \in \Omega \subseteq \mathbb{R} \tag{1}$$

In above, $\mathcal{L}$ is an invertible linear operator, e.g., $\mathcal{L} = \frac{d^k}{dt^k}\big(\bullet\big)$ with k being the highest order of derivative, $\mathcal{R}$ is the remainder of the linear operator, $\mathcal{N}$ is the nonlinear operator, and $g(t)$ is a known source function. We point out that the choice for the linear operator is not unique. For example, with regard to Lane-Emden equation, $\frac{d^2}{dt^2}\big(x(t)\big) + \frac{Q}{t} + \frac{d}{dt}\big(x(t)\big) + g(x) = 0$, one can choose any of the following linear operators: $\mathcal{L} = \frac{d^2}{dt^2}\big(\bullet\big)$ or $t^{-Q}\frac{d}{dt}\big(t^Q\frac{d}{dt}(\bullet)\big)$.

Eq. (1) is accompanied by a set of boundary conditions as ,

$$x^{(0)}(t_0) = \gamma(0), \quad x^{(1)}(t_1) = \gamma(1), \quad x^{(2)}(t_2) = \gamma(2), ..., x^{(k-1)}(t_{k-1}) = \gamma(k-1) \tag{2}$$

In order to restrict the statement of the problem to IVPs[1], it is assumed that ,

$$t_0 = t_1 = t_2 = ... = t_{k-1} \tag{3}$$

In principle, ADM considers an inverse operator as,

$$\mathcal{L}^{-1} = \int_{t_0}^{t}\int_{t_1}^{t}...\int_{t_{k-2}}^{t}\int_{t_{k-1}}^{t}\big(\bullet\big)dt^k \tag{4}$$

The inverse operator transforms Eq.(1) into ,

$$\begin{aligned} &x(t) - x(t_0) - \frac{d}{dt}x(t_0)(t-t_0) - \frac{1}{2!}\frac{d^2}{dt^2}x(t_0)(t-t_0)^2 - \sum_{p=3}^{k-1}\frac{1}{p!}x^{(p)}(t_0)(t-t_0)^p \\ &= \mathcal{L}^{-1}\big(\mathcal{N}[x] + \mathcal{R}[x]\big) + \mathcal{L}^{-1}\big(g(t)\big) \end{aligned} \tag{5}$$

The solution and the nonlinearity are decomposed as,

$$x(t) = x_0(t) + x_1(t) + x_2(t) + ... = \sum_{i=0}^{\infty}x_i(t), \quad \mathcal{N}\big[x(t)\big] = \sum_{i=0}^{\infty}\mathcal{A}_i\big(x(t)\big) \tag{6}$$

In above, $\mathcal{A}_i$ are Adomian terms. To obtain the Adomian terms, we expand $\mathcal{N}\big[x(t)\big]$ about the function $x_0(t)$,

$$\mathcal{N}\big[x(t)\big] = \mathcal{N}\big[x_0\big] + (x-x_0)\frac{d}{dx_0}\mathcal{N}\big[x_0\big] + \frac{(x-x_0)^2}{2!}\frac{d^2}{dx_0^2}\mathcal{N}\big[x_0\big] + \frac{(x-x_0)^3}{3!}\frac{d^3}{dx_0^3}\mathcal{N}\big[x_0\big] + ... \tag{7}$$

Therefore,

$$\mathcal{N}\big[x(t)\big] = \mathcal{N}\big[x_0\big] + \big(\sum_{i=1}^{\infty}x_i\big)\frac{d}{dx_0}\mathcal{N}\big[x_0\big] + \frac{\big(\sum_{i=1}^{\infty}x_i\big)^2}{2!}\frac{d^2}{dx_0^2}\mathcal{N}\big[x_0\big] + \frac{\big(\sum_{i=1}^{\infty}x_i\big)^3}{3!}\frac{d^3}{dx_0^3}\mathcal{N}\big[x_0\big] + ... \tag{8}$$

After some manipulations, we obtain ,

$$\mathcal{N}\big[x(t)\big] = \mathcal{N}\big[x_0\big] + \sum_{n=1}^{\infty}\Big(\sum_{k=1}^{k=n}\frac{1}{k!}\Big(\sum_{i_1,i_2,...,i_k}\delta_{n,i_1+i_2+...+i_k}x_{i_1}x_{i_2}...x_{i_k}\Big)\mathcal{N}^{(k)}\big[x_0\big]\Big) \tag{9}$$

Where,$\delta_{i,j}$is Kronecker delta ,

$$\delta_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{10}$$

[1]It is worth mentioning that ADM can also be adjusted to Boundary Value Problems (BVP). This adjustment is known as Duan-Rach (D-R) ADM (see [42] and [43]). To track a frontier nonlinear decomposition solver, we refer to Optimal Decomposition Method (ODM) recently proposed by Jafarimoghaddam, Soler et al. [44].

Eq. (9) together with Eq.(6) gives ,

$$\mathcal{A}_n = \sum_{k=1}^{k=n} \frac{1}{k!} \Big(\sum_{i_1,i_2,\ldots,i_k} \delta_{n,i_1+i_2+\ldots+i_k} x_{i_1} x_{i_2} \ldots x_{i_k} \Big) \mathcal{N}^{(k)}\big[x_0\big] \tag{11}$$

Therefore,

$$\begin{aligned}
&\mathcal{A}_0\big(x_0(t)\big) = \mathcal{N}\big(x_0\big) \\
&\mathcal{A}_1\big(x_0(t), x_1(t)\big) = x_1 \frac{d}{dx_0} \mathcal{N}\big(x_0\big) \\
&\mathcal{A}_2\big(x_0(t), x_1(t), x_2(t)\big) = x_2 \frac{d}{dx_0} \mathcal{N}\big(x_0\big) + \frac{x_1^2}{2!} \frac{d^2}{dx_0^2} \mathcal{N}\big(x_0\big) \\
&\mathcal{A}_3\big(x_0(t), x_1(t), x_2(t), x_3(t)\big) = x_3 \frac{d}{dx_0} \mathcal{N}\big(x_0\big) + x_1 x_2 \frac{d^2}{dx_0^2} \mathcal{N}\big(x_0\big) + \frac{x_1^3}{3!} \frac{d^3}{dx_0^3} \mathcal{N}\big(x_0\big) \\
&\ldots
\end{aligned} \tag{12}$$

Adomian and Rach proposed[45],

$$\mathcal{A}_n\big(x_0(t), x_1(t), \ldots x_n(t)\big) = \frac{1}{n!} \frac{d^n}{dp^n} \Big[\mathcal{N}\big(\sum_{j=0}^{\infty} x_j p^j \big) \Big]_{p=0} \tag{13}$$

Connecting Eq. (5) and Eq. (6), the recursion scheme becomes,

$$\begin{aligned}
&x_0(t) = x(t_0) + \frac{d}{dt} x(t_0)(t - t_0) + \frac{1}{2!} \frac{d^2}{dt^2} x(t_0)(t - t_0) + \sum_{p=3}^{k-1} \frac{1}{p!} x^{(p)}(t_0)(t - t_0)^p + \mathcal{L}^{-1}\big(g(t)\big) \\
&x_n(t) = \mathcal{L}^{-1}\big(\mathcal{R}\big[x_{n-1}\big]\big) + \mathcal{L}^{-1}\big(\mathcal{A}_{n-1}\big), \quad n \geq 1
\end{aligned} \tag{14}$$

ADM can be shown to diverge in many practical nonlinearities (see e.g., Sec.3.1). To circumvent it, we propose a modified version (SADM).

3 | SEGMENTED ADOMIAN DECOMPOSITION METHOD (SADM)

In SADM, the domain of interest is divided into some intervals,

$$\Omega = \bigcup_{j=1}^{N} \Omega_j\big(t_{j-1}, t_j\big) \tag{15}$$

The recursion scheme in the first interval is obtained by,

$$\begin{aligned}
&x_{0,\Omega_1}(t) = x(t_0) + \sum_{p=1}^{k-1} \frac{1}{p!} x^{(p)}(t_0)\big(t - t_0\big)^p + \mathcal{L}_{\Omega_1}^{-1}\big(g(t)\big) \\
&x_{n,\Omega_1}(t) = \mathcal{L}_{\Omega_1}^{-1}\big(\mathcal{R}\big[x_{n-1,\Omega_1}\big]\big) + \mathcal{L}_{\Omega_1}^{-1}\big(\mathcal{A}_{n-1}\big), \quad n \geq 1
\end{aligned} \tag{16}$$

Therefore, the solution in the first interval is,

$$x_{\Omega_1}(t) = x_{0,\Omega_1} + x_{1,\Omega_1} + \ldots = \sum_{i=0}^{n} x_{i,\Omega_1} \tag{17}$$

For the proceeding intervals,

$$\begin{aligned}
&x_{0,\Omega_j}(t) = x_{\Omega_{j-1}}(t_{j-1}) + \sum_{p=1}^{k-1} \frac{1}{p!} x_{\Omega_{j-1}}^{(p)}(t_{j-1})\big(t - t_{j-1}\big)^p + \mathcal{L}_{\Omega_j}^{-1}\big(g(t)\big) \\
&x_{n,\Omega_j}(t) = \mathcal{L}_{\Omega_j}^{-1}\big(\mathcal{R}\big[x_{n-1,\Omega_j}\big]\big) + \mathcal{L}_{\Omega_j}^{-1}\big(\mathcal{A}_{n-1}\big), \quad n \geq 1
\end{aligned} \tag{18}$$

Thus,

$$x_{\Omega_j}(t) = x_{0,\Omega_j} + x_{1,\Omega_j} + \ldots = \sum_{i=0}^{n} x_{i,\Omega_j}, \quad j \geq 2 \tag{19}$$

The Adomian terms are computed by,

$$\mathcal{A}_n\big(x_{0,\Omega_j}(t), x_{1,\Omega_j}(t), ...x_{n,\Omega_j}(t)\big) = \frac{1}{n!}\frac{d^n}{dp^n}\big[\mathcal{N}\big(\sum_{i=0}^{\infty} x_{i,\Omega_j}p^i\big)\big]_{p=0}, \quad j \geq 1 \tag{20}$$

$x(t_0)$ and $x^{(p)}(t_0)$, $p = 1, 2, ...$, are known form the boundary conditions and n is the order of perturbed fields in each interval.

Theorem 1. The analytic series solution by SADM is convergent on Ω_j if $\mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}$ is a contraction , that is, $\|\mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}\| < 1$.

Proof: We can reconstruct the nonlinearity over the interval Ω_j as,

$$\mathcal{L}_{\Omega_j}\big[x(t)\big] - \hat{\mathcal{N}}_{\Omega_j}\big[x(t)\big] = g(t), \quad \hat{\mathcal{N}} = \mathcal{N} + \mathcal{R} \tag{21}$$

Since, $\mathcal{N}$ is a nonlinear operator, so is $\hat{\mathcal{N}}$. Therefore,

$$\mathcal{L}\big(\sum_{i=0}^{\infty} x_{i,\Omega_j}\big) - \sum_{i=0}^{\infty}\hat{\mathcal{A}}_i = g(t) \tag{22}$$

From Eq. (22),

$$\begin{aligned} x_{0,\Omega_j} &= \mathcal{G}(t) \\ x_{1,\Omega_j} &= \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{A}}_0 \\ x_{2,\Omega_j} &= \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{A}}_1 \\ &... \\ x_{n,\Omega_j} &= \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{A}}_{n-1} \end{aligned} \tag{23}$$

In above, from Eq. (18), $\mathcal{G}(t) = \mathcal{B}(t) + \mathcal{L}_{\Omega_j}^{-1}g(t)$, where, $\mathcal{B}(t)$ is obtained from the boundary conditions.
For a truncated series, we can write ,

$$\hat{\mathcal{N}}\big[X_n\big] = \sum_{i=0}^{n}\hat{\mathcal{A}}_i\big(x_{0,\Omega_j}, ..., x_{i,\Omega_j}\big), \quad X_n = \sum_{i=0}^{n} x_{i,\Omega_j}, \quad S_n = \sum_{i=1}^{n} x_{i,\Omega_j} \tag{24}$$

Therefore,

$$\begin{aligned} &x_{1,\Omega_j} = S_1 = \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{A}}_0\big(x_{0,\Omega_j}\big) = \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}\big(x_{0,\Omega_j}\big) \\ &x_{1,\Omega_j} + x_{2,\Omega_j} = S_2 = \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{A}}_0\big(x_{0,\Omega_j}\big) + \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{A}}_1\big(x_{0,\Omega_j}, x_{1,\Omega_j}\big) = \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}\big(x_{0,\Omega_j} + x_{1,\Omega_j}\big) \\ &... \\ &x_{1,\Omega_j} + x_{2,\Omega_j} + ... + x_{n+1,\Omega_j} = S_{n+1} = \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{A}}_0\big(x_{0,\Omega_j}\big) + \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{A}}_1\big(x_{0,\Omega_j}, x_{1,\Omega_j}\big) + ...+ \\ &\mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{A}}_n\big(x_{0,\Omega_j}, x_{1,\Omega_j}, ..., x_{n,\Omega_j}\big) = \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}\big(x_{0,\Omega_j} + S_n\big) \end{aligned} \tag{25}$$

The recurrent relation, $S_{n+1} = \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}\big(x_{0,\Omega_j} + S_n\big)$,can be written as, $S = \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}\big(x_{0,\Omega_j} + S\big)$. Therefore, S is the fixed point of $\mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}$,[46], and the sequence is convergent, i.e., $\lim_{n\to\infty}\big(S_{n+1} - S_n\big) \to 0$ if $\mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}$ is a contraction , that is, $\|\mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}\| < 1$.■

Theorem 2. Let $\Pi = \Upsilon\big[\Omega_j\big] = \Upsilon\big[t_{j-1}, t_j\big]$, $1 \leq j \leq N$ be the Banach space. Assume that $\mathcal{T}(t, x) = \mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}\big[x\big]$ satisfies Lipschitz condition that is, $\|\mathcal{T}(t, u) - \mathcal{T}(t, v)\| \leq \kappa\|u - v\| \quad \forall u, v \in \Pi$, then, Eq. (21) has only one solution if $\kappa < 1$

Proof: let u and v be two distinct solutions of Eq. (21). Then we can write,

$$\begin{aligned} u &= \mathcal{T}\big(t, u\big) + \mathcal{B}_u(t) + \mathcal{L}_{\Omega_j}^{-1}g(t) \\ v &= \mathcal{T}\big(t, v\big) + \mathcal{B}_v(t) + \mathcal{L}_{\Omega_j}^{-1}g(t) \end{aligned} \tag{26}$$

since $\mathcal{B}_u(t)$ and $\mathcal{B}_v(t)$ are obtained from the boundary conditions, we have, $\mathcal{B}_u(t) = \mathcal{B}_v(t)$. Therefore,

$$\|u - v\| = \|\mathcal{T}(t, u) - \mathcal{T}(t, v)\| \leq \kappa\|u - v\| \tag{27}$$

Since, $\kappa < 1$, then, $u = v$.■

Corollary 1. The operator $\mathcal{L}_{\Omega}^{-1}\hat{\mathcal{N}}$ is not necessarily a contraction. This restrains the applications of the standard ADM. In other words, the standard ADM is convergent if a fixed point theorem holds for $\mathcal{L}_{\Omega}^{-1}\hat{\mathcal{N}}$ over the entire domain. A fixed point theorem exists for continuous monotonic functions such as a corollary on Knaster-Tarski fixed point theorem allocating it to the case of monotonic functions and Kleene's theorem with the functions being additionally continuous (see[47] and[48]). However, the existence of a fixed point theorem does not hold for a general non-monotonic function except for some specific classes. Therefore, if the solution to the nonlinearity can be assumed as a combination of segmented monotonic functions, such that $\|\mathcal{L}_{\Omega_j}^{-1}\hat{\mathcal{N}}\| < 1$, then SADM can be unconditionally convergent.

This corollary indicates the reason for which the standard ADM fails in many practical applications (an analogous comment is also applicable to HAM [2]). This justifies the need for constructing SADM. Similar to the standard ADM, the solutions by SADM are perturbed-analytic; i.e., the solutions are exact near the perturbation origin and a bifurcation appears by moving away from this origin. Therefore, SADM is indeed the practical expediency to analytically tackle the nonlinearities. See the example we present in Section 3.1 to illustrate this feature.

In this paper, two symbolic codes[3] were developed to conveniently calculate the Adomian terms.

3.1 | Elucidating Example

The laminar jet flow over a stationary wall obeys the celebrated Glauert equation[49]. This third order nonlinear differential equation can be formulated in a form resembling the system dynamics,

$$\begin{aligned}
&\frac{d}{dt}x^{(1)} = x^{(2)}\\
&\frac{d}{dt}x^{(2)} = x^{(3)}\\
&\frac{d}{dt}x^{(3)} = -x^{(1)}x^{(3)} - 2x^{(2)}x^{(2)}\\
&x^{(1)}(0) = x^{(2)}(0) = 0, \quad x^{(3)}(0) = \frac{2}{9}, \quad 0 \le t \le \infty
\end{aligned} \tag{28}$$

In above, $x^{(1)}, x^{(2)}$ and $x^{(3)}$ are the state variables. This nonlinear equation has a fast exponentially decaying solution at large t and is highly resistive to admit an analytic series solution. In this respect, the standard ADM is strictly divergent for Eq. (28). Furthermore, the solution provided by HAM, after a careful choice for the auxiliary linear operator, shows a very small admissible radius of convergence in the associated $\hbar$-curve[38]. However, SADM reveals itself as a successful remedy. To give a demonstration of SADM, the second order analytic solution is provided as,

$$\begin{aligned}
&x_{\Omega_j}^{(1)}(t) = a + bt + \frac{c}{2}t^2 - \frac{2b^2+ac}{6}t^3\\
&x_{\Omega_j}^{(2)}(t) = a + ct - \frac{2b^2+ac}{2}t^2 + \frac{2ab^2+a^2c-5bc}{6}t^3\\
&x_{\Omega_j}^{(3)}(t) = c - \big(2b^2+ac\big)t + \frac{2ab^2+a^2c-5bc}{2}t^2 - \frac{a^3c+2a^2b^2-11abc-12b^3+5c^2}{6}t^3\\
&a = x_{\Omega_j}^{(1)}\big(t_{j-1}\big), \quad b = x_{\Omega_j}^{(2)}\big(t_{j-1}\big), \quad c = x_{\Omega_j}^{(3)}\big(t_{j-1}\big), \quad j = 1, ..., N\\
&x_{\Omega_j}^{(1),(2),(3)}\big(t_{j-1}\big) = x_{\Omega_{j-1}}^{(1),(2),(3)}\big(t_{j-1}\big), \quad j = 2, ..., N
\end{aligned} \tag{29}$$

For the sake of clarity, we define an error for the jet component, $x^{(2)}$, as,

$$R = \left| \frac{1}{t_f}\Big(\sum_{j=1}^{N} \int_{\Omega_j} x^{(2)}(t)dt \Big) - \frac{1}{t_f}\int_0^{t_f} \bar{x}^{(2)}dt \right| \tag{30}$$

[2]Roughly speaking, HAM is distinguished by allowing other types of linear operators, or equivalently $\mathcal{L}_{\Omega}^{-1}$, appear. In this respect, HAM may be interpreted in such a way that it searches for a contraction through manipulating $\mathcal{L}_{\Omega}^{-1}$.

[3]Symbolic code 1 is associated with a global analytic solution in terms of recursive Jacobian compositions of the state variables for autonomous system dynamics with a piece-wise constant control vector (See Section 4). Symbolic code 2 (which is a standard form for general non-autonomous systems) employs Eq. (13) in a fast track mode, that is to obtain $\mathcal{A}_n$ based on $x(t) = x_0 + px_1 + ... + p^n x_n$, due to the fact that x_{n+1} does not appear in $\mathcal{A}_n$.

In above,

$$x^{(2)}_{\Omega_j}(t) = x^{(2)}_{0,\Omega_j}(t) + x^{(2)}_{1,\Omega_j}(t) + x^{(2)}_{2,\Omega_j}(t) + ... \tag{31}$$

Moreover, $\bar{x}(t)$ is the closed form solution for the jet[50]. Table 1 shows a comparison between the analytic series solutions of varying perturbed fields with $t_f = 20$ which is an excellent upper bound to fully capture the behavior of the outer flow. The computational time for the symbolic codes for different orders of the perturbed fields is tabulated in Table 2.

It is worth mentioning that in the zeroth order SADM ($P = 0$), there is no consideration of the perturbed fields and it is straightforward to show that this scenario is equivalent to the explicit Euler method at discrete points.

TABLE 1 the quality of the residual error R, Eq. (30), for the elucidating example (3.1)

N	P	P	P	P	P	P	P
	0	1	2	3	4	5	6
15	5.5×10^{26}	1.7×10^{-2}	1.4×10^{-2}	8.9×10^{-3}	2.3×10^{-3}	6.4×10^{-4}	1.3×10^{-4}
20	5.8×10^{57}	1.1×10^{-2}	1.8×10^{-3}	1.1×10^{-3}	3.5×10^{-4}	7.9×10^{-5}	7.6×10^{-6}
30	1.2×10^{138}	2.5×10^{-3}	8.5×10^{-4}	2.5×10^{-4}	6.7×10^{-5}	2.3×10^{-5}	2.1×10^{-6}
40	4.7×10^{178}	8.1×10^{-4}	4.6×10^{-4}	5.8×10^{-5}	2.3×10^{-5}	9.6×10^{-6}	8.9×10^{-7}

TABLE 2 the symbolic computational time for the elucidating example (3.1) in different orders of the perturbed fields, P

P	symbolic code 2 (seconds)	symbolic code 1 (seconds)
0	0.16	0.03
1	0.50	0.04
2	0.68	0.05
3	0.81	0.05
4	0.97	0.06
5	1.15	0.07
6	1.23	0.09

Therefore, compared to the existing analytic nonlinear solvers, SADM seems satisfactory to analytically solve the system dynamics with a good level of accuracy.

4 | PERTURBED-ANALYTIC DIRECT TRANSCRIPTION FOR OPTIMAL CONTROL (PADOC)

Without loss of generality, we state the OCP in Mayer form. For simplification, the state constraints are ignored, although the direct optimization can easily handle them too. Moreover, the control vector is assumed to be piecewise constant which facilitates

understanding of PADOC.

$$\begin{aligned}
&\min_{u\in U} \mathcal{J} = \Phi\big(x(t_f), t_f\big)\\
&s.t.\\
&\frac{d}{dt}\vec{x} = \vec{f}\big(\vec{x}(t), \vec{u}(t)\big)\\
&\vec{g}\big(\vec{u}(t)\big) \le 0\\
&\vec{\varphi}_0\big(\vec{x}(t_0), t_0\big) = 0\\
&\vec{\varphi}_f\big(\vec{x}(t_f), t_f\big) = 0
\end{aligned} \tag{32}$$

Where, $\vec{x} \in \mathbb{R}^L$ is the state vector, $\vec{u} \in \mathbb{R}^K$ is the control vector, $\vec{\varphi}_0 \in \mathbb{R}^P$, $P \le L$ and $\vec{\varphi}_f \in \mathbb{R}^Q$, $Q \le L$ are boundary conditions and $\vec{g} \in \mathbb{R}^O$ is the vector of inequality constraints with $\frac{\partial \vec{g}}{\partial \vec{u}}$ having full rank.

The first challenge is to solve the system dynamics. SADM assumes segments/intervals such that, $\Omega = \bigcup_{j=1}^{n} \Omega_j\big(t_{j-1}, t_j\big)$. The following notations should be clarified beforehand ,

$$\vec{x}_{\Omega_j} = \big(x^{(1)}_{\Omega_j}(t), x^{(2)}_{\Omega_j}(t), ..., x^{(L)}_{\Omega_j}(t)\big)^T \tag{33}$$

$$\vec{x}_{i,\Omega_j} = \big(x^{(1)}_{i,\Omega_j}(t), x^{(2)}_{i,\Omega_j}(t), ..., x^{(L)}_{i,\Omega_j}(t)\big)^T \tag{34}$$

$$\bar{x}_{i,\Omega_j} = \vec{x}^T_{i,\Omega_j} \tag{35}$$

$$\bar{x}^P_{\Omega_j} = \big(\sum_{i=0}^{\infty} p^i x^{(1)}_{i,\Omega_j}, \sum_{i=0}^{\infty} p^i x^{(2)}_{i,\Omega_j}, ..., \sum_{i=0}^{\infty} p^i x^{(L)}_{i,\Omega_j}\big) \tag{36}$$

$$\vec{u}_{\Omega_j} = \big(u^{(1)}_{\Omega_j}(t), u^{(2)}_{\Omega_j}(t), ..., u^{(K)}_{\Omega_j}(t)\big)^T \tag{37}$$

$$\bar{u}_{\Omega_j} = \vec{u}^T_{\Omega_j} \tag{38}$$

$$\vec{\mathcal{A}}_i = \big(\mathcal{A}^{(1)}_i, \mathcal{A}^{(2)}_i, ..., \mathcal{A}^{(L)}_i\big)^T \tag{39}$$

$$J_a\big(\vec{\Xi}(\vec{x})\big) = \frac{\partial\big(\Xi^{(1)}, ..., \Xi^{(L)}\big)}{\partial\big(x^{(1)}, ..., x^{(L)}\big)} \tag{40}$$

In above, $\big(\bullet\big)^T$ is the matrix transpose.

The solution for each element of the state vector is defined as,

$$x^{(l)}_{\Omega_j}(t) = x^{(l)}_{0,\Omega_j} + x^{(l)}_{1,\Omega_j} + ... + x^{(l)}_{n,\Omega_j} \tag{41}$$

In above, the analytic-perturbed solutions are obtained by,

$$\begin{aligned}
x^{(l)}_{1,\Omega_j} &= \mathcal{L}^{-1}_{\Omega_j}\big(\mathcal{A}^{(l)}_0\big)\\
x^{(l)}_{2,\Omega_j} &= \mathcal{L}^{-1}_{\Omega_j}\big(\mathcal{A}^{(l)}_1\big)\\
&...\\
x^{(l)}_{n,\Omega_j} &= \mathcal{L}^{-1}_{\Omega_j}\big(\mathcal{A}^{(l)}_{n-1}\big)
\end{aligned} \tag{42}$$

Following the decomposition method, we now proceed with computing Adomian terms of any order.

Adomian term of the zeroth order is computed as,

$$\mathcal{A}_0^{(l)} = f^{(l)}\left(\bar{x}_{0,\Omega_j}; \bar{u}_{\Omega_j}\right) \tag{43}$$

Therefore, the Adomian vector of the zeroth order is,

$$\vec{\mathcal{A}}_0 = \vec{f}\left(\bar{x}_{0,\Omega_j}; \bar{u}_{\Omega_j}\right) \tag{44}$$

For the Adomian term of the first order, we have,

$$\begin{aligned} &\mathcal{A}_1^{(l)} = \frac{d}{dp}\left[f^{(l)}\left(x_{0,\Omega_j}^{(1)} + px_{1,\Omega_j}^{(1)} + ...; x_{0,\Omega_j}^{(2)} + px_{1,\Omega_j}^{(2)} + ...; x_{0,\Omega_j}^{(l)} + px_{1,\Omega_j}^{(l)} + ...; \bar{u}_{\Omega_j}\right)\right]_{p=0} \\ &\rightarrow \mathcal{A}_1^{(l)} = \frac{d}{dp}\left[f^{(l)}\left(\bar{x}_{\Omega_j}^{p}; \bar{u}_{\Omega_j}\right)\right]_{p=0} \end{aligned} \tag{45}$$

Therefore,

$$\mathcal{A}_1^{(l)} = \frac{\partial f^{(l)}}{\partial x_{0,\Omega_j}^{(1)}} x_{1,\Omega_j}^{(1)} + \frac{\partial f^{(l)}}{\partial x_{0,\Omega_j}^{(2)}} x_{1,\Omega_j}^{(2)} + ... + \frac{\partial f^{(l)}}{\partial x_{0,\Omega_j}^{(L)}} x_{1,\Omega_j}^{(L)} \tag{46}$$

Eventually, we can write the Adomain vector of the first order as,

$$\vec{\mathcal{A}}_1 = \frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}} \vec{x}_{1,\Omega_j} \tag{47}$$

Proceeding, as before, for the Adomian vector of the second order,

$$\vec{\mathcal{A}}_2 = \frac{1}{2!}\frac{d^2}{dp^2}\left[\vec{f}\left(\bar{x}_{\Omega_j}^{p}; \bar{u}_{\Omega_j}\right)\right]_{p=0} = \frac{1}{2!}\frac{d}{dp}\left[\frac{\partial \vec{f}\left(\bar{x}_{\Omega_j}^{p}; \bar{u}_{\Omega_j}\right)}{\partial\left(\bar{x}_{\Omega_j}^{p}\right)^T}\frac{d}{dp}\left(\bar{x}_{\Omega_j}^{p}\right)^T\right]_{p=0} \tag{48}$$

In an expanded form, it is equivalent to,

$$\begin{aligned} \vec{\mathcal{A}}_2 = &\Big[\frac{1}{2!}\frac{\partial \vec{f}\left(\bar{x}_{\Omega_j}^{p}; \bar{u}_{\Omega_j}\right)}{\partial\left(\bar{x}_{\Omega_j}^{p}\right)^T}\frac{d^2}{dp^2}\left(\bar{x}_{\Omega_j}^{p}\right)^T + \frac{1}{2!}\sum_{l=1}^{L}\Big[\frac{\partial}{\partial\sum_{i=0}^{\infty}p^i x_{i,\Omega_j}^{(l)}}\frac{\partial \vec{f}\left(\bar{x}_{\Omega_j}^{p}; \bar{u}_{\Omega_j}\right)}{\partial\left(\bar{x}_{\Omega_j}^{p}\right)^T}\frac{d}{dp}\sum_{i=0}^{\infty}p^i x_{i,\Omega_j}^{(l)}\Big] \\ &\frac{d}{dp}\left(\bar{x}_{\Omega_j}^{p}\right)^T\Big]_{p=0} \end{aligned} \tag{49}$$

After simplification, we obtain,

$$\vec{\mathcal{A}}_2 = \frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}\vec{x}_{2,\Omega_j} + \frac{1}{2!}\sum_{l=1}^{L}\Big[\frac{\partial}{\partial x_{0,\Omega_j}^{(l)}}\frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}x_{1,\Omega_j}^{(l)}\Big]\vec{x}_{1,\Omega_j} \tag{50}$$

Following an analogous procedure, for the third order Adomian vector, we get,

$$\begin{aligned} \vec{\mathcal{A}}_3 = &2\times 2!\sum_{l=1}^{L}\Big[\frac{\partial}{\partial x_{0,\Omega_j}^{(l)}}\frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}x_{1,\Omega_j}^{(l)}\Big]\vec{x}_{2,\Omega_j} + 3!\frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}\vec{x}_{3,\Omega_j} \\ &+\sum_{l=1}^{L}\Big[\sum_{l=1}^{L}\frac{\partial}{\partial x_{0,\Omega_j}^{(l)}}\frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}x_{1,\Omega_j}^{(l)}x_{1,\Omega_j}^{(l)} + 2!\frac{\partial}{\partial x_{0,\Omega_j}^{(l)}}\frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}x_{2,\Omega_j}^{(l)}\Big]\vec{x}_{1,\Omega_j} \end{aligned} \tag{51}$$

Theorem 3. for the system dynamics $\frac{d}{dt}\vec{x} = \vec{f}\left(\vec{x}(t), \vec{u}(t)\right)$ with the initial boundary conditions $\vec{\varphi}_0\left(\vec{x}(t_0), t_0\right) = 0$, and $\vec{u}(t) = \bigcup_{j=1}^{N}\vec{u}_{\Omega_j} = \bigcup_{j=1}^{N}\left(u_{\Omega_j}^{(1)}, ..., u_{\Omega_j}^{(L)}\right)^T = \bigcup_{j=1}^{N}\left(c_{\Omega_j}^{(1)}, ..., c_{\Omega_j}^{(L)}\right)^T$, SADM over $\Omega_j : \Omega = \bigcup_{j=1}^{N}\Omega_j\left(t_{j-1}, t_j\right)$ is reduced to the following

recursive Jacobian compositions' form ,

$$\begin{aligned}
&\vec{x}_{\Omega_j}(t) = \vec{x}(t_{j-1}) + \sum_{i=0}^{n\to\infty} \vec{\mathcal{F}}_i\big(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\big)\frac{\big(t-t_{j-1}\big)^{i+1}}{\big(i+1\big)!} \\
&= \vec{x}(t_{j-1}) + \vec{\mathcal{F}}_0\big(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\big)\big(t-t_{j-1}\big) + ... + \vec{\mathcal{F}}_n\big(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\big)\frac{\big(t-t_{j-1}\big)^{n+1}}{\big(n+1\big)!} \\
&\vec{\mathcal{F}}_0\big(\vec{x}, \vec{c}_{\Omega_j}\big) = \vec{f}\big(\vec{x}, \vec{c}_{\Omega_j}\big) \\
&\vec{\mathcal{F}}_1\big(\vec{x}, \vec{c}_{\Omega_j}\big) = J_a\Big(\vec{\mathcal{F}}_0\big(\vec{x}, \vec{c}_{\Omega_j}\big)\Big)\vec{\mathcal{F}}_0\big(\vec{x}, \vec{c}_{\Omega_j}\big) \\
&\vec{\mathcal{F}}_2\big(\vec{x}, \vec{c}_{\Omega_j}\big) = J_a\Big(\vec{\mathcal{F}}_1\big(\vec{x}, \vec{c}_{\Omega_j}\big)\Big)\vec{\mathcal{F}}_0\big(\vec{x}, \vec{c}_{\Omega_j}\big) \\
&... \\
&\vec{\mathcal{F}}_n\big(\vec{x}, \vec{c}_{\Omega_j}\big) = J_a\Big(\vec{\mathcal{F}}_{n-1}\big(\vec{x}, \vec{c}_{\Omega_j}\big)\Big)\vec{\mathcal{F}}_0\big(\vec{x}, \vec{c}_{\Omega_j}\big)
\end{aligned} \tag{52}$$

$\vec{\mathcal{F}}_i$ is the global perturbed vector of i^{th} order and $\vec{c}_{\Omega_j} = \big(c^{(1)}_{\Omega_j}, ..., c^{(L)}_{\Omega_j}\big)^T$ is the piecewise constant control vector.

The proof to Theorem 3 is included in an appendix.

5 | PRIMAL-DUAL COMPLETENESS OF PADOC

Here, we prove that the multipliers associated with PADOC in NLP correspond to a backward analytic solution of the continuous adjoint/costate equations in PMP. In addition, it is substantiated that the associated cost functional, in Bolza form, is transcribed into an exact analytic form.

For the OCP defined in 32, the augmented cost is,

$$\bar{\mathcal{J}} = \Phi\big(\vec{x}(t_f), t_f\big) + \vec{\nu}_0^T\vec{\varphi}_0 + \vec{\nu}_f^T\vec{\varphi}_f + \int_{t_0}^{t_f}\Big[\vec{\lambda}^T(t)\Big(\vec{f}\big(\vec{x}(t), \vec{u}(t)\big) - \frac{d}{dt}\vec{x}(t)\Big) + \vec{\mu}^T(t)\vec{g}(t)\Big]dt \tag{53}$$

In above, $\vec{\lambda}(t)$, costate functions, and $\vec{\mu}(t)$ are the time-dependent multipliers, $\vec{\nu}_0$ and $\vec{\nu}_f$ are constant multipliers to involve the initial and final boundary constraints.

The augmented Hamiltonian associated with Eq. (53) is,

$$\mathcal{H}\big(\vec{x}, \vec{u}, \vec{\lambda}\big) = \vec{\lambda}^T\vec{f} + \vec{\mu}^T\vec{g} \tag{54}$$

The necessary optimality conditions are,

$$\frac{\partial\mathcal{H}}{\partial\vec{u}} = 0, \quad \vec{\mu}^T\vec{g} = 0, \quad \vec{\mu} \geq 0 \tag{55}$$

The costate dynamics together with the transversality conditions are,

$$\frac{d}{dt}\vec{\lambda}^T = -\frac{\partial\mathcal{H}}{\partial\vec{x}} \tag{56}$$

$$\vec{\lambda}(t_0) = -\Big[\frac{\partial\vec{\varphi}_0}{\partial\vec{x}(t_0)}\Big]^T\vec{\nu}_0 \tag{57}$$

$$\vec{\lambda}(t_f) = \frac{\partial\Phi}{\partial\vec{x}(t_f)} + \Big[\frac{\partial\vec{\varphi}_f}{\partial\vec{x}(t_f)}\Big]^T\vec{\nu}_f \tag{58}$$

$$\mathcal{H}(t_f) = -\Big[\frac{\partial\Phi}{\partial t_f} + \vec{\nu}_f^T\frac{\partial\vec{\varphi}_f}{\partial t_f}\Big] \tag{59}$$

The cost functional is discretized at the nodes where the analytic solution is partitioned. Therefore, we write the discretized version of the cost functional (the Lagrangian) as,

$$\hat{\mathcal{J}} = \Phi\big(\vec{x}(t_N), t_N\big) + \vec{\nu}_0^T \vec{\varphi}_0 + \vec{\nu}_N^T \vec{\varphi}_N + \sum_{j=1}^{N} \vec{\lambda}_j^T \vec{\Lambda}_j + \vec{\mu}_j^T \vec{g}\big(\vec{u}(t_j)\big) \tag{60}$$

where $\vec{\Lambda}_j$ are the nodal state constraints,

$$\vec{\Lambda}_j = \vec{x}_{\Omega_j}(t_j) - \vec{x}(t_j) \tag{61}$$

Hence, we have,

$$\vec{\Lambda}_j = \vec{x}(t_{j-1}) - \vec{x}(t_j) + \sum_{n=0}^{\infty} \vec{\mathcal{F}}_n\big(\vec{x}(t_{j-1}), \vec{u}_{\Omega_j}\big) \frac{\big(t_j - t_{j-1}\big)^{n+1}}{(n+1)!} \tag{62}$$

By defining, $h = t_j - t_{j-1}$ the above expression is equivalent to,

$$\vec{\Lambda}_j = \vec{x}_{j-1} - \vec{x}_j + \sum_{n=0}^{\infty} \vec{\mathcal{F}}_n\big(\vec{x}_{j-1}, \vec{u}_j\big) \frac{h^{n+1}}{(n+1)!} \tag{63}$$

In above, $\vec{\mathcal{F}}_n\big(\vec{x}, \vec{u}_{\Omega_j}\big)$ is computed by theorem (3).
The KKT first-order necessary conditions for the parametric OCP are,

$$\begin{aligned} &\frac{\partial \hat{\mathcal{J}}}{\partial \vec{x}_k} = 0, \quad \frac{\partial \hat{\mathcal{J}}}{\partial \vec{u}_k} = 0, \quad \frac{\partial \hat{\mathcal{J}}}{\partial t_N} = 0 \\ &\vec{\mu}_k^T \vec{g}_k = 0, \quad \vec{\mu}_k \geq 0, \quad k = 1, ..., N \end{aligned} \tag{64}$$

For the interior points, i.e., $k = 1, ..., N-1$ we have,

$$\frac{\partial \hat{\mathcal{J}}}{\partial \vec{x}_k} = \sum_{j=1}^{N} \vec{\lambda}_j^T \frac{\partial \vec{\Lambda}_j}{\partial \vec{x}_k} = 0 \tag{65}$$

$\vec{x}_k$ appears only in the adjacent nodal state constraints. Therefore,

$$\vec{\lambda}_k^T \frac{\partial \vec{\Lambda}_k}{\partial \vec{x}_k} + \vec{\lambda}_{k+1}^T \frac{\partial \vec{\Lambda}_{k+1}}{\partial \vec{x}_k} = 0 \tag{66}$$

From Eq. (63), we can obtain,

$$\frac{\partial \vec{\Lambda}_k}{\partial \vec{x}_k} = -I \tag{67}$$

$$\frac{\partial \vec{\Lambda}_{k+1}}{\partial \vec{x}_k} = I + \sum_{n=0}^{\infty} \frac{\partial}{\partial \vec{x}_k} \vec{\mathcal{F}}_n\big(\vec{x}_k, \vec{u}_{k+1}\big) \frac{h^{n+1}}{(n+1)!} \tag{68}$$

In above, I is the identity matrix. Substituting Eq. (67) and Eq. (68) into Eq. (66),

$$-\vec{\lambda}_k^T + \vec{\lambda}_{k+1}^T + \vec{\lambda}_{k+1}^T \sum_{n=0}^{\infty} \frac{\partial}{\partial \vec{x}_k} \vec{\mathcal{F}}_n\big(\vec{x}_k, \vec{u}_{k+1}\big) \frac{h^{n+1}}{(n+1)!} = 0 \tag{69}$$

Next, we show that Eq. (69) corresponds to a backward analytic solution of the system of the adjoint equations by SADM. The system of adjoint equations can be expressed as:

$$\frac{d}{dt} \vec{\lambda}^T = -\vec{\lambda}^T \frac{\partial \vec{f}\big(\vec{x}(t), \vec{c}(t)\big)}{\partial \vec{x}} \tag{70}$$

Similar to the state equations, the analytic solution for the adjoint equations is developed over an arbitrary interval Ω_j. By choosing $\mathcal{L}_{\Omega_j}^{-1} = \int_t^{t_j} \big(\bullet\big) dt$, we get ,

$$\vec{\lambda}_{\Omega_j}^T(t) = \vec{\lambda}^T(t_j) + \mathcal{L}_{\Omega_j}^{-1}\Big[\vec{\lambda}^T \frac{\partial \vec{f}\big(\vec{x}(t), \vec{c}(t)\big)}{\partial \vec{x}}\Big] \tag{71}$$

By expansion,

$$\vec{\lambda}^T_{\Omega_j}(t) = \vec{\lambda}^T_{0,\Omega_j} + \vec{\lambda}^T_{1,\Omega_j} + ... + \vec{\lambda}^T_{n,\Omega_j} \tag{72}$$

Therefore, we can write,

$$\begin{aligned} \vec{\lambda}^T_{0,\Omega_j} &= \vec{\lambda}^T(t_j) \\ \vec{\lambda}^T_{1,\Omega_j} &= \mathcal{L}^{-1}_{\Omega_j}\vec{\mathcal{B}}^T_0 \\ &... \\ \vec{\lambda}^T_{n,\Omega_j} &= \mathcal{L}^{-1}_{\Omega_j}\vec{\mathcal{B}}^T_{n-1} \end{aligned} \tag{73}$$

Where, $\vec{\mathcal{B}}^T_n$ is the transpose Adomian vector of the n^{th} order for the system of adjoint equations. The zeroth term is,

$$\vec{\mathcal{B}}^T_0 = \vec{\lambda}^T_{0,\Omega_j}\frac{\partial \vec{f}\left(\bar{x}_{0,\Omega_j};\bar{u}_{\Omega_j}\right)}{\partial \vec{x}} \tag{74}$$

From above, we can note,

$$\vec{\mathcal{B}}^T_0 = \vec{\lambda}^T_{0,\Omega_j} J_a(\vec{f})_{\vec{x}=\vec{x}_j} = \vec{\lambda}^T(t_j) J_a(\vec{\mathcal{F}}_0)_{\vec{x}=\vec{x}_j} \tag{75}$$

Therefore,

$$\vec{\lambda}^T_{1,\Omega_j} = \mathcal{L}^{-1}_{\Omega_j}\vec{\mathcal{B}}^T_0 = \vec{\lambda}^T(t_j) J_a(\vec{\mathcal{F}}_0)_{\vec{x}=\vec{x}_j}\left(t_j - t\right) \tag{76}$$

For $\vec{\mathcal{B}}^T_1$ we write,

$$\vec{\mathcal{B}}^T_1 = \frac{d}{dp}\left[\left(\vec{\lambda}^T_{0,\Omega_j} + p\vec{\lambda}^T_{1,\Omega_j} + ... + p^n\vec{\lambda}^T_{n,\Omega_j}\right)\frac{\partial \vec{f}\left(\bar{x}^P_{\Omega_j};\bar{u}_{\Omega_j}\right)}{\vec{x}^P_{\Omega_j}}\right]_{p=0} \tag{77}$$

By simplification, we obtain,

$$\vec{\mathcal{B}}^T_1 = \vec{\lambda}^T_{0,\Omega_j}\sum_{l=1}^{L}\left[\frac{\partial}{\partial x^{(l)}_{0,\Omega_j}}\frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}x^{(l)}_{1,\Omega_j}\right] + \vec{\lambda}^T_{1,\Omega_j}\frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}} \tag{78}$$

We note that the integration is backward; therefore, by substituting $x^{(l)}_{1,\Omega_j}$ and $\vec{\lambda}^T_{1,\Omega_j}$ in Eq. (78), we get,

$$\vec{\mathcal{B}}^T_1 = \vec{\lambda}^T_{0,\Omega_j}\left(\sum_{l=1}^{L}\left[\frac{\partial}{\partial x^{(l)}_{0,\Omega_j}}\frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}f^{(l)}\left(\bar{x}_{0,\Omega_j};\bar{u}_{\Omega_j}\right)\right] + \frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}\frac{\partial \vec{f}}{\partial \vec{x}_{0,\Omega_j}}\right)\left(t_j - t\right) \tag{79}$$

Thus,

$$\vec{\mathcal{B}}^T_1 = \vec{\lambda}^T(t_j) J_a(J_a(\vec{f})\vec{f})_{\vec{x}=\vec{x}_j}\left(t_j - t\right) = \vec{\lambda}^T(t_j) J_a(\vec{\mathcal{F}}_1)_{\vec{x}=\vec{x}_j}\left(t_j - t\right) \tag{80}$$

Having $\vec{\mathcal{B}}^T_1$ found,

$$\vec{\lambda}^T_{2,\Omega_j}(t) = \mathcal{L}^{-1}_{\Omega_j}\vec{\mathcal{B}}_1 = \frac{\left(t_j - t\right)^2}{2!}\vec{\lambda}^T(t_j) J_a(\vec{\mathcal{F}}_1)_{\vec{x}=\vec{x}_j} \tag{81}$$

By induction, we obtain,

$$\vec{\lambda}^T_{n,\Omega_j}(t) = \mathcal{L}^{-1}_{\Omega_j}\vec{\mathcal{B}}_{n-1} = \frac{\left(t_j - t\right)^n}{n!}\vec{\lambda}^T(t_j) J_a(\vec{\mathcal{F}}_{n-1})_{\vec{x}=\vec{x}_j} \tag{82}$$

Therefore, we can set down the Global Analytic solution for the system of adjoint equations as,

$$\vec{\lambda}^T_{\Omega_j}(t) = \vec{\lambda}^T(t_j) + \vec{\lambda}^T(t_j)\sum_{i=0}^{n\to\infty} J_a\left(\vec{\mathcal{F}}_i(\vec{x},\vec{c}_{\Omega_j})\right)_{\vec{x}=\vec{x}_j}\frac{\left(t_j - t\right)^{i+1}}{(i+1)!} \tag{83}$$

Therefore, in the nodal constraints, we can write,

$$-\vec{\lambda}^T_k + \vec{\lambda}^T_{k+1} + \vec{\lambda}^T_{k+1}\sum_{n=0}^{\infty} J_a\left(\vec{\mathcal{F}}_n(\vec{x},\vec{c}_{\Omega_j})\right)_{\vec{x}=\vec{x}_{k+1}}\frac{h^{n+1}}{(n+1)!} = 0 \tag{84}$$

Therefore, we can conclude that the multipliers (associated with nodal state constraints) in the direct optimization by PADOC correspond to a backward analytic solution of the adjoint equations. The interesting point is that the discrete multipliers by the direct optimization can be propagated analytically through the entire interval Ω_j by Eq. (83). For the boundary conditions, we proceed, as before, by differentiation with respect to $\vec{x}(t_N)$,

$$\frac{\partial \Phi\left(\vec{x}(t_N), t_N\right)}{\partial \vec{x}(t_N)} + \left[\frac{\partial \vec{\varphi}_f\left(\vec{x}(t_N), t_N\right)}{\partial \vec{x}(t_N)}\right]^T \vec{\tilde{\nu}}_f - \vec{\tilde{\lambda}}_N = 0 \tag{85}$$

This is equivalent to the boundary conditions for the adjoint system.
For the optimality condition, we have ,

$$\begin{aligned} &\frac{\partial \hat{\mathcal{J}}}{\partial \vec{u}_k} = \vec{\tilde{\lambda}}_k^T \frac{\partial \vec{\Lambda}_k}{\partial \vec{u}_k} + \vec{\tilde{\mu}}_k^T \frac{\partial \vec{g}(\vec{u}_k)}{\partial \vec{u}_k} \\ &\vec{\tilde{\mu}}_k^T \vec{g}_k = 0, \quad \vec{\tilde{\mu}}_k \geq 0, \quad k = 1, ..., N \end{aligned} \tag{86}$$

Therefore,

$$\vec{\tilde{\mu}}_k^T \frac{\partial \vec{g}(\vec{u}_k)}{\partial \vec{u}_k} + \vec{\tilde{\lambda}}_k^T \sum_{n=0}^{\infty} \frac{\partial}{\partial \vec{u}_k} \vec{\mathcal{F}}_n\left(\vec{x}_{k-1}, \vec{u}_k\right) \frac{h^{n+1}}{(n+1)!} = 0 \tag{87}$$

The next step is to show that Eq. (87) corresponds to the optimality condition defined by the discretized Hamiltonian,

$$\frac{\partial \mathcal{H}\left(\vec{x}_k, \vec{\lambda}_k, \vec{u}_k\right)}{\partial \vec{u}_k} = \vec{\lambda}_k^T \frac{\partial \vec{f}}{\partial \vec{u}_k} + \vec{\mu}_k^T \frac{\partial \vec{g}}{\partial \vec{u}_k} = 0, \quad \vec{\mu}_k^T \vec{g}_k = 0, \quad \vec{\mu}_k \geq 0 \tag{88}$$

To this end, we write,

$$\frac{\partial \vec{f}}{\partial \vec{u}_k} = \frac{\partial}{\partial \vec{u}_k} \frac{d}{dt} \vec{x} = \frac{\partial}{\partial \vec{u}_k} \sum_{i=0}^{n \to \infty} \vec{\mathcal{F}}_i\left(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\right) \frac{\left(t - t_{j-1}\right)^i}{i!} \tag{89}$$

Dividing Eq. (87) by h,

$$\frac{\vec{\tilde{\mu}}_k^T}{h} \frac{\partial \vec{g}(\vec{u}_k)}{\partial \vec{u}_k} + \vec{\tilde{\lambda}}_k^T \frac{\partial}{\partial \vec{u}_k} \bar{f} = 0 \tag{90}$$

Where,

$$\bar{f} = \frac{1}{h} \int_{t_{k-1}}^{t_k} \vec{f}\left(t, \vec{c}_{\Omega_j}\right) dt \tag{91}$$

Therefore, $\vec{f}$ is replaced with a pure analytic average over the interval Ω_j which is the most accurate scenario. Moreover,

$$\frac{\vec{\tilde{\mu}}_k^T}{h} = \vec{\mu}_k^T \tag{92}$$

The equivalence of the transversality condition follows an analogous fashion.

Theorem 4. assume that for a specific OCP, $\min_{\vec{u}_k} \hat{\mathcal{J}}^{P^{(1)}} = \min_{\vec{u}_m} \hat{\mathcal{J}}^{P^{(2)}}$. Therefore, $\vec{\tilde{\lambda}}_k^{P^{(1)}} = \vec{\tilde{\lambda}}_k^{P^{(2)}}$.

$$P^{(1)} : \begin{cases} \frac{\partial \hat{\mathcal{J}}}{\partial \vec{x}_k} = 0, \quad \frac{\partial \hat{\mathcal{J}}}{\partial \vec{u}_k} = 0, \quad \frac{\partial \hat{\mathcal{J}}}{\partial t_N} = 0 \\ \vec{\tilde{\mu}}_k^T \vec{g}_k = 0, \quad \vec{\tilde{\mu}}_k \geq 0 \\ k = 1, ..., N \end{cases} \tag{93}$$

$$P^{(2)} : \begin{cases} \frac{\partial \hat{\mathcal{J}}}{\partial \vec{x}_k} = 0, \quad \frac{\partial \hat{\mathcal{J}}}{\partial \vec{u}_m} = 0, \quad \frac{\partial \hat{\mathcal{J}}}{\partial t_N} = 0 \\ \vec{\tilde{\mu}}_m^T \vec{g}_m = 0, \quad \vec{\tilde{\mu}}_m \geq 0 \\ k = 1, ..., N \\ m = 1, ..., M \\ M < N \end{cases} \tag{94}$$

Proof: Under the assumption of the uniqueness of the solution for the specific OCP, we must have $\vec{\bar{\lambda}}_k^{P^{(1)}} = \vec{\bar{\lambda}}_k^{*,P^{(1)}} = \vec{\bar{\lambda}}_k^{*}$. Since, $\min_{\vec{u}_k} \hat{\mathcal{J}}^{P^{(1)}} = \min_{\vec{u}_m} \hat{\mathcal{J}}^{P^{(2)}}$, the following holds, $\vec{\bar{\lambda}}_k^{P^{(2)}} = \vec{\bar{\lambda}}_k^{*,P^{(2)}} = \vec{\bar{\lambda}}_k^{*}$. Therefore, $\vec{\bar{\lambda}}_k^{P^{(1)}} = \vec{\bar{\lambda}}_k^{P^{(2)}}$.■

Corollary 2. assume that theorem (4) holds for a specific unconstrained OCP with $M = 1$. Therefore, from Eq. (90),

$$P^{(1)} : \begin{cases} \vec{\bar{\lambda}}_1^T \frac{\partial}{\partial \vec{u}_1} \bar{f} = 0 \\ \vec{\bar{\lambda}}_2^T \frac{\partial}{\partial \vec{u}_2} \bar{f} = 0 \\ ... \\ \vec{\bar{\lambda}}_N^T \frac{\partial}{\partial \vec{u}_N} \bar{f} = 0 \end{cases} \tag{95}$$

$$P^{(2)} : \left\{ \left(\vec{\bar{\lambda}}_1^T + \vec{\bar{\lambda}}_2^T + ... + \vec{\bar{\lambda}}_N^T \right) \frac{\partial}{\partial \vec{u}_M} \bar{f} = 0 \right. \tag{96}$$

From Eq. (95) and Eq. (96),

$$\left(\vec{\bar{\lambda}}_1^T + \vec{\bar{\lambda}}_2^T + ... + \vec{\bar{\lambda}}_N^T \right) \frac{\partial}{\partial \vec{u}_M} \bar{f} = \vec{\bar{\lambda}}_1^T \frac{\partial}{\partial \vec{u}_1} \bar{f} + \vec{\bar{\lambda}}_2^T \frac{\partial}{\partial \vec{u}_2} \bar{f} + ... + \vec{\bar{\lambda}}_N^T \frac{\partial}{\partial \vec{u}_N} \bar{f} \tag{97}$$

This means that $\frac{\partial}{\partial \vec{u}_M} \bar{f}$ is the average of $\frac{\partial}{\partial \vec{u}_k} \bar{f}$ with the weights $\vec{\bar{\lambda}}_k^T$. Therefore, it is reasonable to look at it as an average NonLinear Programming (aNLP).

Corollary 3. we must have $M = N$ beforehand, to instate theorem (4). However, in practice, as M increases, $\min_{\vec{u}_k} \hat{\mathcal{J}}^{P^{(1)}}$ approaches drastically to $\min_{\vec{u}_m} \hat{\mathcal{J}}^{P^{(2)}}$. Therefore, for a specific OCP, there exists M such that theorem (4) holds with an extreme accuracy.

Remark 1. for regular OCPs, aNLP leads to a quite sparse nonlinear optimization. More specifically, with $M << N$, PADOC gives multipliers that can be propagated in a series form over the sub-intervals. Next, by positioning multipliers over the nodal state constraints (whose multipliers appear in the optimality condition, $\frac{d\mathcal{H}}{du} = 0$), the staircase optimal solutions from aNLP can be converted to the distributed optimal solutions.

In several examples, we show that even $M = 1$ provides a very good estimation of the continuous control law.

6 | CONSTRUCTION OF THE PADOC ALGORITHM

In this paper, the key elements of PADOC are SADM and aNLP. In this section, we present the PADOC algorithm, launched upon SADM and aNLP.

In the PADOC algorithm, the optimal structure is termed regular if the optimal control does not contain singular/bang extremals. Otherwise, the optimal structure is termed irregular.

The PADOC algorithm is designed to handle both regular and irregular OCPs. In short, first, the general optimal structure in the form of staircase function is detected. Next, the outcome is fed to either PADOC handling regular OCP or PADOC handling irregular OCP.

PADOC handling regular OCP exploits aNLP theorem (4 and remark 1).

PADOC handling irregular OCP exploits a discontinuity function to identify the number of discontinuities as additional decision variables. In the case that Hamiltonian is linear on the control vector, the singular extremal is expressed by a series of a truncated order in PADOC handling irregular OCP.

Now, we present PADOC algorithm for a generic OCP in Bolza form. A generic Bolza OCP can be pictured as[51]:

$$\begin{aligned}
&\min_{u\in U} \mathcal{J} = \Phi\big(x(t_0), x(t_f), t_0, t_f\big) + \int_{t_0}^{t_f} \pounds\big(x(t), u(t), t\big)dt \\
&s.t. \\
&\frac{d}{dt}x = f\big(x(t), u(t), t\big) \\
&g\big(u(t), x(t), t\big) \leq 0 \\
&\varphi\big(x(t_0), x(t_f), t_0, t_f\big) \leq 0
\end{aligned} \tag{98}$$

The scalar $t \in [t_0, t_f]$ is time, $x \in W^{1,\infty}\big([t_0, t_f], \mathbb{R}^n\big)$ is the state vector and $u \in L^{\infty}\big([t_0, t_f], \mathbb{R}^p\big)$ is the control vector. $\mathcal{J}$ is the cost encompassing a boundary objective $\Phi : \mathbb{R}^{2n+2} \to \mathbb{R}$ and an integral objective $\pounds : \mathbb{R}^n \times \mathbb{R}^p \times [t_0, t_f] \to \mathbb{R}$. Dynamics, path/control and boundary constraints are $f : \mathbb{R}^n \times \mathbb{R}^p \times [t_0, t_f] \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^p \times [t_0, t_f] \to \mathbb{R}^s$ and $\varphi : \mathbb{R}^{2n+2} \to \mathbb{R}^r$ respectively.

Next, we employ PADOC to analytically transcribe the continuous OCP into a NLP. It should be pointed out that our analysis in the previous section is qualitatively portable to the present section. Therefore, for the sake of conciseness, presenting the parametric OCP for the above Bolza form is omitted. The only remaining point is that the integral functional is computed analytically by PADOC.

Eventually, if Mangasarian Fromovitz Constraint Qualification (MFCQ) holds at the local optimal solution, then, there exist unique multipliers that satisfy KKT conditions and the solution is reachable through parameter optimization.

6.1 | PADOC algorithm

Step 1: the control vector is assumed as piecewise constant; i.e., $u(t) = \bigcup_{j=1}^{M} u_{\Omega_j} = \bigcup_{j=1}^{M} \big(c_{\Omega_j}^{(1)}, c_{\Omega_j}^{(2)}, ..., c_{\Omega_j}^{(p)}\big)^T$ over $\Omega_j : \Omega = \bigcup_{j=1}^{M} \Omega_j(t_{j-1}, t_j)$.

Step 2: we solve the state equations over $\Omega_i : \Omega = \bigcup_{i=1}^{N} \Omega_i(t_{i-1}, t_i)$, $N \geq M$ with a truncated order of the perturbed fields. It should be pointed out that for autonomous systems, the Adomian terms are computed by symbolic code 1 (associated with theorem 3) and for the non-autonomous systems we apply symbolic code 2.

Step 3: we solve the resulting NLP and extract the staircase solution. In the case of regular OCP, we go to **Step 4**; otherwise, to **Step 9**.

Step 4 (**PADOC handling regular OCP**): applying aNLP (theorem 4 and remark 1) we extract the optimal solution over Ω_i by turning the staircase solution to a semi-continuous one.

Step 5: The optimal solution due to **Step 4** is applied as an initial guess over Ω_i and the resulting NLP is solved.

The following steps (**Step 6**, **Step 7** and **Step 8**) are designed to extract analytic expressions for the state and control variables over the minimum possible number of intervals.

Step 6: we find the *hp* adaptation by relaxing N and evaluating the global squared residual error for the system dynamics defined as,

$$\begin{aligned}
\bar{R} &= \sum_{k=1}^{n} R^{(k)} \\
R &= \frac{1}{N}\sum_{i=1}^{N} \frac{1}{t_i - t_{i-1}} \int_{\Omega_i} \Big[\frac{d}{dt}x_{\Omega_i} - f\big(x_{\Omega_i}, \bar{u}\big)\Big]^2 dt \\
\bar{u} &= \frac{1}{t_i - t_{i-1}} \int_{\Omega_i} u(t)dt
\end{aligned} \tag{99}$$

We start with $N := 1$ and increase the order of the perturbed fields, P , to see if the residual error meets a predefined tolerance, i.e., $\bar{R} < \epsilon$. If not, the sequence is repeated for $N = 2, 3, \cdots$. The outcome is N^* and P^*. Therefore, the intervals, for both states and controls, are modified as: $\Omega_k : \Omega = \bigcup_{k=1}^{N^*} \Omega_k(t_{k-1}, t_k)$.

Step 7: we express the control vector in a series of a truncated order defined as,

$$u_{\Omega_k}(t) = \sum_{n=0}^{\bar{n}} \sum_{m=0}^{\bar{m}} C_{n,m} t^m e^{-nt} \tag{100}$$

The above series is the resulting base functions of an exponential-type linear operator in HAM with a rapid rate of convergence (see[36] and[37].)

Step 8: the resulting NLP, initialized by the solution from **Step 5**, is solved and the analytic expressions for the state and control vectors are extracted.

Step 9 (**PADOC handling irregular OCP**): we identify the possible discontinuities by the jump function $[u](t) = u(t^+) - u(t^-)$ (for a better resolution of the jump function, we can employ the methods in[52] and[53]).

Step 10: it is important to note beforehand that the SADM solution for the system dynamics ensures that the state vector does not violate the possible state constraints in any time instants within the entire time domain. The same argument should hold for the control vector. To this end, for an OCP with constraints on the control we can either,

1. Express the control vector in terms of Bernstein polynomials which satisfy the property of a convex hull, i.e., $conv(x_v) := \left\{ \sum_{v=0}^{n} x_v \lambda_v | \left(\forall v : \lambda_v \geq 0\right) \wedge \sum_{v=0}^{n} \lambda_v = 1 \right\}$. See[51] for more information.
2. Treat the number of possible discontinuities as extra decision variables in the frame of NLP and express the control, outside the bang region, by Eq. (100).

Here, we have followed the latter scenario. For K number of discontinuities, the additional decision variables are $\tilde{t}_k, k = 1, ..., K$ and the time domain is partitioned as,

$$\Omega = \bigcup_{k=1}^{K} \Omega_k \left(\tilde{t}_{k-1}, \tilde{t}_k\right) \tag{101}$$

Step 11: *hp* adaptation is checked over the subdomain Ω_k. For each extremal Ω_k, SADM is applied over $\Delta\tilde{t}_k$,

$$\Delta\tilde{t}_k = \frac{\tilde{t}_k - \tilde{t}_{k-1}}{N_k} = t_j - t_{j-1}, \quad j = 1 + \sum_{i=1}^{k-1} N_i, ..., \sum_{i=1}^{k} N_i, \quad k = 1, ..., K \tag{102}$$

The global residual error can be rewritten as,

$$\begin{aligned} \bar{R}_k &= \sum_{p=1}^{n} R_k^{(p)} \\ R_k &= \frac{1}{N_k} \sum_{j=1+\sum_{i=1}^{k-1} N_i}^{\sum_{i=1}^{k} N_i} \frac{1}{t_j - t_{j-1}} \int_{\Omega_j} \left[\frac{d}{dt} x_{\Omega_j} - f\left(x_{\Omega_j}, \bar{u}\right) \right]^2 dt \\ \bar{u} &= \frac{1}{t_j - t_{j-1}} \int_{\Omega_j} u(t) dt \end{aligned} \tag{103}$$

We start with $N_k := 1$ and increase the order of perturbed fields and check if $\bar{R}_k \leq \epsilon$. If not, we set $N_k = 2, 3, ...$ and repeat this step. The outcome is P_k^* and N_k^*.

Step 12: over the bang extremals (Ω_k^b), the control vector is set as piecewise constant. As for the singular extremals (Ω_k^s), we use the series function, Eq. (100), up to a truncated order.

Step 13: we solve the resulting NLP initialized by the staircase solution in **Step 3**.

7 | ILLUSTRATIVE EXAMPLES

This section is devoted to some clarifying examples. For the sake of brevity, PADOC is inspected in more details throughout the first four examples (7.1, 7.2, 7.3, 7.4). However, for the next examples (7.5, 7.6), we have mainly displayed the quality of the optimal solutions by PADOC and comparisons wherever applicable.

In this research, interior-point/barrier algorithm of the NLP solver **fmincon** from MATLAB® Optimization Toolbox was adopted as the optimization module. The codes within the present paper were developed in MATLAB®, R2020b, on a laptop, configured by Intel(R) Core(TM) i7 (10TH GEN)-10750H CPU @ 2.60GHz.

7.1 | Orbital Transfer

The OCP of transferring a rocket from an initial orbit to a target orbit in a fixed time is expressed as[41],

$$
\begin{aligned}
&\min_{\epsilon(t)} \mathcal{J} = \Phi(t_f) = -\big[\frac{1}{2}\big(u^2(t_f) + v^2(t_f)\big) - \frac{1}{r(t_f)}\big] \\
&s.t. \\
&\frac{d}{dt}r = u \\
&\frac{d}{dt}\theta = \frac{v}{r} \\
&\frac{d}{dt}u = \frac{v^2}{r} - \frac{1}{r^2} + A\sin\epsilon(t) \\
&\frac{d}{dt}v = -\frac{uv}{r} + A\cos\epsilon(t) \\
&A = 0.01, \quad r(0) = 1.1, \quad \theta(0) = 0, \quad u(0) = 0, \quad v(0) = \frac{1}{\sqrt{1.1}}, \quad t_f = 50
\end{aligned}
\tag{104}
$$

The control variable is the thrust steering angle measured from the local horizontal $\epsilon(t)$.

Interestingly, we can show, by order of magnitude analysis, that the solution to Eq. (104) is in close proximity to the solution with $\epsilon(t) = 0$. Therefore, the objective is almost unaffected by the control (to be compared later). Moreover, in a mathematical point of view, this is a triple state OCP since $\theta(t)$ is isolated.

For this OCP, Hamiltonian can be written as,

$$
\mathcal{H} = \lambda_r u + \lambda_u\Big(\frac{v^2}{r} - \frac{1}{r^2} + A\sin\epsilon\Big) + \lambda_v\Big(-\frac{uv}{r} + A\cos\epsilon\Big) \tag{105}
$$

Therefore, PMP for the optimality condition, $\frac{\partial\mathcal{H}}{\partial\epsilon} = 0$, gives,

$$
\epsilon(t) = \tan^{-1}\frac{\lambda_u(t)}{\lambda_v(t)} \tag{106}
$$

SADM over Ω_j reads,

$$
\begin{aligned}
r(t) &= r_{t=t_{j-1}} + u_{t=t_{j-1}}t + \big(A\sin\epsilon_{\Omega_j} - \frac{1}{r^2} + \frac{v^2}{r}\big)_{t=t_{j-1}}t^2 + ... \\
u(t) &= u_{t=t_{j-1}} + \big(A\sin\epsilon_{\Omega_j} - \frac{1}{r^2} + \frac{v^2}{r}\big)_{t=t_{j-1}}t \\
&+ \Big(\frac{2u}{r^3} - \frac{uv^2}{r^2} + \frac{2v}{r}\big(A\sin\epsilon_{\Omega_j} - \frac{uv}{r}\big)\Big)_{t=t_{j-1}}t^2 + ... \\
v(t) &= v_{t=t_{j-1}} + \big(A\sin\epsilon_{\Omega_j} - \frac{uv}{r}\big)_{t=t_{j-1}}t \\
&+ \Big(\frac{u^2v}{r^2} - \frac{v}{r}\big(A\sin\epsilon_{\Omega_j} - \frac{1}{r^2} + \frac{v^2}{r}\big) - \frac{u}{r}\big(A\sin\epsilon_{\Omega_j} - \frac{uv}{r}\big)\Big)_{t=t_{j-1}}t^2 + ...
\end{aligned}
\tag{107}
$$

For this OCP, aNLP can be initiated by setting $M = 1$. This is equivalent to $\epsilon(t) = \bar{\epsilon}$. SADM of the 1^{st} order was confirmed to be quite accurate and stable over 50 nodes. Therefore, we take $N = 50$.

From Eq. (106), the optimal control is a function of $\lambda_u(t)$ and $\lambda_v(t)$. Therefore, it is reasonable to ignore positioning the nodal state constraints for $\lambda_r(t)$ in aNLP.

For 50 nodes, 49 segments, and $M = 1$, the size of the aNLP is 99 variables. Next, we solve the aNLP by the optimization module and extract multipliers $\tilde{\lambda}_u$, and $\tilde{\lambda}_v$. Eventually, we plug these multipliers into Eq. (106) and the optimal control law with the resolution of the state nodes (50 nodes) is extracted. Fig. 1-a shows the performance of aNLP with $M = 1, 5, 10$. For the initial guess, we used a constant steering of 0.001 rad.

Clearly, aNLP, even with $M = 1$, provides a very good topological approximation of the continuous control. Table (3) displays the performance of the optimization module on independent aNLP solutions.

TABLE 3 Performance of the optimization module on aNLP in example (7.1)

M	iterations	Optimization Time (seconds)	Optimal Cost
1	3	0.17	0.095841
2	7	0.29	0.095734
5	12	0.38	0.095184
10	16	0.72	0.095132
50	44	3.2	0.095118

In fact, for $M = 1$, the actual aNLP has only one decision variable and the existence of the nodal state constraints is not essential and NLP can indirectly retain the concept of the multipliers associated with the nodal state constraints. In this respect, for the aNLP with $M = 1$, we have performed a direct optimization over the only existent decision variable, $\epsilon(t) = \bar{\epsilon}$. The main aspect of this case is to provide a qualitative visual on the influence of different transcription methods on the Hessian search direction, accuracy and the effective domain for the initial guesses. Fig. 1-b shows the behavior of the objective as a function of $\epsilon(t) = \bar{\epsilon}$ in applying second order SADM in the PADOC algorithm. Fig. 1-c shows an analogous behavior in a stable Euler method.

In the end, in the case that theorem (4) holds, there seems to be a complementary theorem that relates the topological local optima from the univariate ring (Fig. 1-b and 1-c) to a multivariate function.

The optimal states from the PADOC algorithm are shown in Fig. 1-d and 1-e where we can verify that the optimal solution is in close proximity to the solution without having a control. For the case of $\epsilon(t) = 0$, we find $\Phi(t_f) = 0.096247$.

Table 4 shows the global squared residual error (Eq. 99) for SADM of a varying order to manifest the hp-adaptive quality for this OCP.

TABLE 4 the hp-adaptive quality, Eq.(99), for example (7.1)

N	P 0	P 1	P 2	P 3	P 4	P 5
20	3.8×10^{3}	3.6×10^{-1}	9.7×10^{-4}	2.4×10^{-4}	8.4×10^{-5}	9.6×10^{-6}
30	2.5×10^{19}	5.8×10^{-3}	2.9×10^{-4}	6.5×10^{-5}	1.2×10^{-5}	3.3×10^{-6}
50	3.8×10^{2}	8.6×10^{-4}	7.6×10^{-5}	9×10^{-6}	3×10^{-7}	2×10^{-8}

7.2 | Van der Pol Oscillator

The problem is defined as[54],

$$\begin{aligned}
&\min_{u(t)} \mathcal{J} = \frac{1}{2}\int_0^2 \big(x_1^2(t) + x_2^2(t) + u^2(t)\big)dt \\
&s.t. \\
&\frac{d}{dt}x_1 = x_2 \\
&\frac{d}{dt}x_2 = -x_1 + x_2(1 - x_1^2) + u(t) \\
&x_1(0) = 1, \quad x_2(0) = 0
\end{aligned} \tag{108}$$

The Hamiltonian for this problem is,

$$\mathcal{H} = \frac{1}{2}\big(x_1^2 + x_2^2 + u^2\big) + \lambda_{x_1} x_2 + \lambda_{x_2}\big(-x_1 + x_2(1 - x_1^2) + u\big) \tag{109}$$

Therefore, form PMP ($\frac{\partial \mathcal{H}}{\partial u} = 0$),

$$u(t) = -\lambda_{x_2}(t) \tag{110}$$

For this OCP, we used SADM of the first order over 40 nodal state constrains ($N = 40$) with $M = 1, 2, 5$. The multipliers were extracted from the optimization module and together with Eq. (110), the optimal control was found (see Fig. 2-a). Indeed, a very good agreement is verified. It should be mentioned that the nodal state constraints were distributed over $x_2(t)$ since the optimal control has no dependency on λ_{x_1}. Table 5 shows the *hp*-adaptive quality for SADM (Eq. 99). Following the PADOC algorithm (**Step 6-8**), we were able to extract high order optimal series solutions for the state equations as well as highly accurate piecewise interpolants for the optimal control. The minimum cost by PADOC is $\mathcal{J} = 1.04285$. Table 6 shows a comparison between the semi-analytic solution reported in[54] and PADOC. Fig. 2-b exhibits the optimal series solutions by PADOC for the state functions. Table 7 shows the performance of the optimization module on aNLP with $M = 1, 2, 5$.

TABLE 5 the *hp*-adaptive quality, Eq.(99), for example 7.2

N	P 0	P 1	P 2	P 3	P 4
2	3.7×10^{-1}	3.2×10^{-3}	1.6×10^{-3}	3.6×10^{-4}	1.2×10^{-4}
3	3.5×10^{-1}	1.1×10^{-3}	6.5×10^{-4}	1.3×10^{-4}	3.4×10^{-5}
4	3.1×10^{-1}	9.1×10^{-4}	3.1×10^{-4}	5.4×10^{-5}	8.6×10^{-6}
5	2.3×10^{-1}	6.3×10^{-4}	1.8×10^{-4}	2.5×10^{-5}	5×10^{-7}

TABLE 6 comparison between PADOC and Pade-HPM[54] for the optimal control $u(t)$ in example 7.2

t	PADOC	Pade-HPM
0	0.010914	–0.011719
0.4	0.517903	0.514115
0.8	0.721478	0.720252
1.2	0.644546	0.645724
1.6	0.370995	0.372690
2	0.009343	0.000775

TABLE 7 Performance of the optimization module on aNLP in example (7.2)

M	iterations	Optimization Time (seconds)	Optimal Cost
1	4	0.12	1.10339
2	8	0.19	1.10205
5	18	0.32	1.05747

7.3 | Brachistochrone Problem

This is among the first appearing OCPs and was solved by Johann Bernoulli by calculus of variations. Brachistochrone OCP can be defined as[55],

$$\begin{aligned}
&\min_{u(t)} \mathcal{J} = t_f \\
&s.t. \\
&\frac{d}{dt}x = v\sin u(t) \\
&\frac{d}{dt}y = \cos u(t) \\
&\frac{d}{dt}v = g\cos u(t) \\
&x(0) = y(0) = v(0) = 0, \quad x(t_f) = y(t_f) = 2
\end{aligned} \tag{111}$$

Let us first consider $M = 1$. The solution of the system dynamics by SADM gives the exact closed form solution. In other words, since the higher order perturbed fields become zero in the Adomian recursion scheme, the induction of the close form solution is immediate. Therefore, the closed form solution by SADM reads ,

$$\begin{aligned}
x(t) &= \frac{g}{4}\sin(2\bar{u})t^2 \\
y(t) &= \frac{g}{2}\cos^2(\bar{u})t^2 \\
v(t) &= g\cos(\bar{u})t
\end{aligned} \tag{112}$$

For $M = 1$, the corresponding aNLP, can be put into the following form,

$$\hat{\mathcal{J}} = t_f + \tilde{v}_{f_x}\left(\frac{g}{4}\sin(2\bar{u})t_f^2 - 2\right) + \tilde{v}_{f_y}\left(\frac{g}{2}\cos^2(\bar{u})t_f^2 - 2\right) \tag{113}$$

Therefore, the optimization module attempts to solve,

$$\begin{aligned}
&\frac{\partial\hat{\mathcal{J}}}{\partial t_f} = 1 + \tilde{v}_{f_x}\frac{g}{2}\sin(2\bar{u})t_f + \tilde{v}_{f_y}g\cos^2(\bar{u})t_f = 0 \\
&\frac{\partial\hat{\mathcal{J}}}{\partial\bar{u}} = \tilde{v}_{f_x}\frac{g}{2}\cos(2\bar{u})t_f^2 - \tilde{v}_{f_y}\frac{g}{2}\sin(2\bar{u})t_f^2 = 0 \\
&\frac{g}{4}\sin(2\bar{u})t_f^2 - 2 = 0 \\
&\frac{g}{2}\cos^2(\bar{u})t_f^2 - 2 = 0
\end{aligned} \tag{114}$$

The close form solution reads ,

$$\bar{u} = \frac{\pi}{4}, \quad t_f = \sqrt{\frac{8}{g}}, \quad \tilde{v}_{f_y} = 0, \quad \tilde{v}_{f_x} = -\frac{2}{\sqrt{8g}} \tag{115}$$

The hp adaptation was found with $\bar{R} = 0$ upon $P = 1$ and $N = 1$. In addition, it was found that $u(t) = at$ best describes the exact optimal control.

On processing the SADM, we found that Adomian terms of the second order forward are all zero. Therefore, the SADM of the 1^{st} order is the exact close form solution. Consequently, given $u(t) = at$, the close form solution by SADM becomes,

$$\begin{aligned} x(t) &= \frac{g}{a}\big(\frac{t}{2} - \frac{\sin(2at)}{4a}\big) \\ y(t) &= \frac{g}{2a^2}\sin^2(at) \\ v(t) &= \frac{g}{a}\sin(at) \end{aligned} \tag{116}$$

In this sense, the augmented cost is,

$$\hat{\mathcal{J}} = t_f + \tilde{v}_{f_x}\Big(\frac{gt_f}{2a} - \frac{g\sin(2at_f)}{4a^2} - 2\Big) + \tilde{v}_{f_y}\big(\frac{g}{2a^2}\sin^2(at) - 2\big) \tag{117}$$

In this situation, the solution by the optimization module for t_f and a corresponds to the solution of the following trigonometric algebraic equations,

$$\begin{aligned} &\frac{gt_f}{2a} - \frac{g\sin(2at_f)}{4a^2} - 2 = 0 \\ &\frac{g}{2a^2}\sin^2(at_f) - 2 = 0 \end{aligned} \tag{118}$$

The close form solution reads,

$$\begin{aligned} &\sin\Big(\big(\sqrt{\frac{g}{a^2} - 4} + 2\big)\frac{2a^2}{g}\Big) = \frac{2a}{\sqrt{g}} \\ &t_f = \big(\sqrt{\frac{g}{a^2} - 4} + 2\big)\frac{2a}{g} \end{aligned} \tag{119}$$

In numerics, the optimal solutions are $a = 1.4629977$ and $t_f = 0.8243387$ which is comparable to the aNLP result (with $M = 1$), $t_f = 0.9030473$.

7.4 | Aly Problem

This linear-quadratic regulator OCP is expressed as[32]:

$$\begin{aligned} &\min_{u(t)} \mathcal{J} = \frac{1}{2}\int_0^5 x_1^2(t) + x_2^2(t)dt \\ &s.t. \\ &\frac{d}{dt}x_1 = x_2 \\ &\frac{d}{dt}x_2 = u(t) \\ &|u(t)| \leq 1 \end{aligned} \tag{120}$$

Through the PADOC algorithm, this problem was solved over 2 segments; $(0, t_d)$ and $(t_d, 5)$. t_d is the identified discontinuity. We found that the Adomian terms of the second order forward were all zero. Therefore, the optimal solution by PADOC for this

OCP is in close form. The optimal solution by PADOC is as follows ,

$$
\begin{aligned}
&t \in (0, t_d)\\
&u(t) = -1\\
&x_1(t) = -\frac{1}{2}t^2 + t\\
&x_2(t) = 1 - t\\
&t \in (t_d, 5)\\
&u(t) = ae^{-t} + b\\
&x_1(t) = ae^{-t} + \frac{b}{2}t^2 + \left(1 - t_d - bt_d + ae^{-t_d}\right)t + t_d - \frac{1}{2}t_d^2\\
&x_2(t) = -ae^{-t} + bt + ae^{-t_d} - bt_d - t_d + 1
\end{aligned}
\tag{121}
$$

The decision variables were optimally identified as; $t_d \approx 1.405714$, $a \approx 1.495751$, $b \approx 0.012781$. Eventually, the optimal cost was computed as, $\mathcal{J} \approx 0.364513$. The switching time is comparable to that in[56] $t_d \approx 1.413764$. The optimal solutions are graphed in Fig. 3-a and 3-b.

7.5 | Maximum Range of a Hang Glider

This OCP is stated as[29],

$$
\begin{aligned}
&\min_{c_L(t)} \mathcal{J} = -x(t_f)\\
&\frac{d}{dt}x = v_x\\
&\frac{d}{dt}y = v_y\\
&\frac{d}{dt}v_x = \frac{1}{m}\Big(- L\sin(\eta) - D\cos(\eta)\Big)\\
&\frac{d}{dt}v_y = \frac{1}{m}\Big(L\cos(\eta) - D\sin(\eta) - W\Big)\\
&0 \le c_L(t) \le 1.4\\
&x(0) = 0, \quad y(0) = 1000, \quad v_x(0) = 13.23, \quad v_y(0) = -1.288\\
&y(t_f) = 900, \quad v_x(t_f) = 13.23, \quad v_y(t_f) = -1.288\\
&\eta = \tan^{-1}\left(\frac{v_y - u_a(x)}{v_x}\right), \quad v_r = \sqrt{v_x^2 + \left(v_y - u_a(x)\right)^2}\\
&L = \frac{1}{2}\rho c_L S v_r^2, \quad D = \frac{1}{2}\rho c_D\left(c_L\right) S v_r^2, \quad W = mg\\
&c_D\left(c_L\right) = c_{D_0} + kc_L^2, \quad u_a(x) = u_{a,max}(x)e^{-\left(\frac{x}{R} - 2.5\right)^2}\left(1 - \left(\frac{x}{R} - 2.5\right)^2\right)^2\\
&u_{a,max} = 2.5, \quad R = 100, \quad c_{D_0} = 0.034, \quad k = 0.069662, \quad m = 100\\
&S = 14, \quad \rho = 1.13, \quad g = 9.81
\end{aligned}
\tag{122}
$$

The values are given in SI unit. The authors in[29] have reported that the convergence cannot be reached for the above OCP in its complete form upon applying a direct collocation method of a cubic Lobatto type. For this reason, they combined direct and indirect approaches together with parameter continuation technique to solve this OCP. On trying PADOC, The *hp* adaptation was reached for 1^{st} order (above about 120 nodes), 2^{nd} order (above about 80 nodes) and 3^{rd} order (above about 50 nodes). It is noteworthy mentioning that there was no sensitivity by PADOC to an initial guess within a practical range. The maximum range computed is, $x(t_f) = 1247.947903$ with the flight duration, $t_f = 98.409791$. The computed quantities agree very well with the report in[29], that is $x(t_f) = 1247.60$ and $t_f = 98.380$. Fig. 4-a shows a comparison between the optimal control by PADOC complete solution compared with the optimal control by aNLP (without interference of the Hamiltonian) with 30 segments, 150 nodal state constraints and SADM of the second order. Fig. 4-b, 4-c, 4-d and 4-e manifest the optimal states by PADOC complete solution. Table 8 shows the time performance of the optimization module with $M = 2, 3, 10, 30$ and second order SADM. It is noteworthy mentioning that in principle, the number of the decision variables needs to exceed the number of

boundary equations. In the case of equivalence, the aNLP turns into a shooting-like scenario, that is to find the decision variables such that the boundary equations are satisfied. For $M = 2$, the aNLP contains 2 segments for $c_L(t)$ plus t_f. Since the OCP has 3 boundary equations, this case coincides with a shooting scenario with no actual decision variables in the sense of optimization. The solutions tabulated in Table 8 are independent with a fixed initial guess. We continued the PADOC algorithm taking into account the aNLP solution with $M = 30$ as the initial guess and the final solution was reached in about $26.3s$.

TABLE 8 Performance of the optimization module on aNLP in example (7.5)

M	iterations	Optimization Time (seconds)	Flight Duration (seconds)	Optimal Cost (meters)
2	5	0.13	90.53366	1199.22208
3	9	0.26	93.58561	1208.61251
10	34	0.95	94.80555	1219.70867
30	69	8.4	98.03995	1244.43072

7.6 | Minimum Time Aircraft Trajectory in Climbing Phase

The original system dynamics governing an aircraft in its climbing phase contains variables with different time scales, mass $m(t)$ and air slop $\gamma(t)$. A normal remedy is to apply singular perturbation (for more details, see [57]). Under a singular perturbation analysis, the system dynamics becomes as follows. Thrust and fuel flow are modeled by Base of Aircraft DAta (BADA) model. In addition, air density is approximated by International Standard Atmospheric (ISA) model.

$$
\begin{aligned}
&\min_{\gamma(t)} \mathcal{J} = t_f \\
&\frac{d}{dt}h = v\gamma(t) \\
&\frac{d}{dt}v = \frac{T(h)}{m} - \frac{1}{2}\rho(h)S\frac{v^2}{m}\Big(C_{D_1} + C_{D_2}\big(\frac{2mg}{\rho(h)Sv^2}\big)^2\Big) - g\gamma(t) \\
&\frac{d}{dt}m = -C_s(v)T(h) \\
&\gamma_{min} \le \gamma(t) \le 0.262 \\
&h(0) = 3480, \quad v(0) = 151.67, \quad m(0) = 69000 \\
&h(t_f) = 9144, \quad v(t_f) = 191, \quad m(t_f) = 68100 \\
&T(h) = C_{T_1}\big(1 - \frac{h}{C_{T_2}} + h^2 C_{T_3}\big), \quad C_s(v) = C_{s_1}\big(1 + \frac{v}{C_{s_2}}\big) \\
&P(h) = P_0\big(\frac{\Theta(h)}{\Theta_0}\big)^{\frac{g}{\beta R}}, \quad \Theta(h) = \Theta_0 - \beta h, \quad \rho(h) = \frac{P(h)}{R\Theta(h)} \\
&S = 122.6, \quad g = 9.81, \quad C_{T_1} = 141040, \quad C_{T_2} = 14909.9 \\
&C_{T_3} = 6.997e^{-10} \quad C_{D_1} = 0.0242, \quad C_{D_2} = 0.0469, \quad C_{s_1} = 1.055e^{-5} \\
&C_{s_2} = 441.54 \quad R = 287.058, \quad \Theta_0 = 288.15, \quad \beta = 0.0065, \quad P_0 = 101325
\end{aligned}
\tag{123}
$$

This singular OCP has been solved in [57] for $\gamma_{min} = -0.262$. Here, we have relaxed γ_{min} to analyze the performance of PADOC in a varying control bound. The PADOC algorithm confirmed a bang-singular-bang solution for this OCP. Next, we partitioned the domain by the global residual error as instructed through the PADOC algorithm. The hp adaptation was confirmed as (with the $\bar{R}$ well below 10^{-6}), one segment over $(0, t_{d_1})$, 5 segments over (t_{d_1}, t_{d_2}) and 1 segment over (t_{d_2}, t_f), each segment with the perturbed fields of the 4^{th} order. Next, we used the interpolant series, Eq. (100), for the singular arc and solved the resulting NLP. For the case of $\gamma_{min} = -0.262$, Fig. 5-a shows a comparison between the report in [57] ($t_f^* = 644.2$), aNLP with $M = 50$ ($t_f^* = 644.52101$) and the PADOC complete solution ($t_f^* = 642.34312$). Fig. 5-b displays the optimal controls for $\gamma_{min} = -0.262$ to $\gamma_{min} = 0.022$ with the step, $\delta\gamma_{min} = 0.01$. Fig. 5-c betrays t_f^* as a function of γ_{min} which shows a marginal affectibility. Fig.

5-d, 5-e and 5-f exhibit the optimal solutions for the state variables. Table 9 depicts the time performance of the optimization module with $M = 2, 5, 10, 50$ and the second order SADM. The final solution was reached in about $8.3s$ on using the results from the aNLP with $M = 50$ as the initial guess.

TABLE 9 Performance of the optimization module on aNLP in example (7.6)

M	iterations	Optimization Time (seconds)	Minimum Time (seconds)
2	7	0.14	658.10901
5	28	0.92	651.79756
10	67	1.8	647.40787
50	113	6.6	644.52101

8 | CONCLUSION

There exist various numerical integration and collocation schemes to solve the system dynamics and to form an NLP and only some of these methods are complete. Reviewing the literature, it appears that the existing transcription methods, have been introduced as alternatives with limited studies on the aspects of the superiority of one over another. Therefore, the ultimate goal will be to introduce PADOC as a unified framework for direct trajectory optimization.

Nevertheless, there has been no general analytic transcription method in the literature until now. In short, the reason lies behind the fact that the area of research belonging to the analytic nonlinear solvers is, in some respects, immature and the so-far known analytic methods are successful only on some types of nonlinearity and they do not guarantee convergence for a wide class of nonlinear problems. This gap was replenished by introducing a powerful analytic nonlinear solver, SADM. This new analytic solver together with aNLP were deployed to form PADOC which is a complete method since it allows an order-preserving map between the dualized problem and the discretized problem.

PADOC was run over many OCPs and its robustness, computational efficiency and reliability were confirmed. Here, some of the distinct attributes of PADOC were manifested through the examples. Specifically, Ex.7.1 (an orbital transfer problem) and Ex.7.2 (the Van der Pol oscillator) were mainly devoted to the role of aNLP and how it reduces the decision variables while retaining the accuracy. In Ex. 7.3 (Brachistochrone problem) and Ex.7.4 (Aly problem) we mainly showed the potentiality of PADOC to extract nearly close form optimal solutions. Eventually, through Ex.7.5 (maximum range of a hang glider) and Ex.7.6 (minimum time aircraft trajectory in climbing phase) we demonstrated the performance of PADOC in more complex arenas with highly accurate outcomes.

The purpose of this research was mainly to introduce PADOC not only as an algorithm but also as a new prospective method to absorb the attention of the research community.

In view of the present authors, the potential extensions to PADOC will be 1) to scrutinize the sparsity of the search direction matrix under PADOC compared to other transcription methods; 2) to explore other linear operators, leading to topologically-different perturbed fields and to analyze the CMP criterion; 3) to extend aNLP for constrained and singular OCPs as a general bridge between the differential theory and the variational theory through Fixed Point Property (FPP); 4) to compare the optimal results by PADOC with other transcription methods in terms of accuracy and the computational time for practical applications.

References

1. Itik M, Salamci MU, Banks SP. Optimal control of drug therapy in cancer treatment. *Nonlinear Analysis: Theory, Methods Applications* 2009; 71(12): e1473-e1486. doi: https://doi.org/10.1016/j.na.2009.01.214

2. Gao X, Teo KL, Duan GR. An optimal control approach to spacecraft rendezvous on elliptical orbit. *Optimal Control Applications and Methods* 2015; 36(2): 158-178. doi: https://doi.org/10.1002/oca.2108

3. Soler M, Kamgarpour M, Lloret J, Lygeros J. A Hybrid Optimal Control Approach to Fuel-Efficient Aircraft Conflict Avoidance. *IEEE Trans. Intell. Transp. Syst.* 2016; 17(7): 1826–1838. doi: 10.1109/TITS.2015.2510824

4. Morante D, Sanjurjo Rivo M, Soler M. Multi-Objective Low-Thrust Interplanetary Trajectory Optimization Based on Generalized Logarithmic Spirals. *Journal of Guidance, Control, and Dynamics* 2019; 42(3): 476-490. doi: 10.2514/1.G003702

5. García-Heras J, Soler M, González-Arribas D, et al. Robust flight planning impact assessment considering convective phenomena. *Transportation Research Part C: Emerging Technologies* 2021; 123: 102968. doi: https://doi.org/10.1016/j.trc.2021.102968

6. Wei S, Zefran M, DeCarlo RA. Optimal control of robotic systems with logical constraints: Application to UAV path planning. In: ; 2008: 176-181

7. Drag P, Styczen K, Kwiatkowska M, Szczurek A. *A Review on the Direct and Indirect Methods for Solving Optimal Control Problems with Differential-Algebraic Constraints*: 91–105; Cham: Springer International Publishing . 2016

8. Reeger J. A comparison of transcription techniques for the optimal control of the International Space Station. In: ; 2009

9. Ross IM, Fahroo F. A Perspective on Methods for Trajectory Optimization. *AIAA/AAS Astrodynamics Specialist Conference and Exhibit* 2012; https://arc.aiaa.org/doi/pdf/10.2514/6.2002-4727. doi: 10.2514/6.2002-4727

10. Ross IM. (2005). A historical introduction to the convector mapping principle.. *In Proceedings of Astrodynamics Specialists Conference. Naval Postgraduate School (US).*

11. Ross M, Fahroo F. Discrete verification of necessary conditions for switched nonlinear optimal control systems. In: . 2. ; 2004: 1610-1615 vol.2

12. Ross IM, Gong Q, Karpenko M, Proulx RJ. Scaling and Balancing for High-Performance Computation of Optimal Controls. *Journal of Guidance, Control, and Dynamics* 2018; 41(10): 2086-2097. doi: 10.2514/1.G003382

13. Tao T. *(2010). An epsilon of room, I: real analysis (Vol. 1). American Mathematical Soc..*

14. Ross IM. A Direct Shooting Method is Equivalent to an Indirect Method. ; 2021: https://arxiv.org/abs/2003.02418v2.

15. Rao AV. (2009). A survey of numerical methods for optimal control. Advances in the Astronautical Sciences, 135(1), 497-528.

16. Betts JT. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Second Edition*. Society for Industrial and Applied Mathematics. second ed. 2010

17. Teo K, Goh C, Wong K. A unified computational approach to optimal control problems;1991. http://hdl.handle.net/20.500.11937/24319.

18. Jennings L, Fisher M, Teo K, Goh C. MISER3:Solving optimal control problems—an update. *Advances in Engineering Software and Workstations* 1991; 13(4): 190-196. doi: https://doi.org/10.1016/0961-3552(91)90016-W

19. Hargraves C, Paris S. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics* 1987; 10(4): 338-342. doi: 10.2514/3.20223

20. Eide HWRA. Modified Legendre–Gauss–Radau Collocation Method for Optimal Control Problems with Nonsmooth Solutions. *J Optim Theory Appl* 1987; 191: 600–633.

21. Fahroo F, Ross IM. Direct Trajectory Optimization by a Chebyshev Pseudospectral Method. *Journal of Guidance, Control, and Dynamics* 2002; 25(1): 160-166. doi: 10.2514/2.4862

22. S K. *Analysis and formulation of a class of complex dynamic optimization problems*. PhD thesis. Carnegie Mellon University, Pittsburgh, PA, 2007.

23. Schlegel M, Marquardt W. Direct Sequential Dynamic Optimization with Automatic Switching Structure Detection. *IFAC Proceedings* 2004; 37(9): 419-424. doi: 10.1016/s1474–6670(17)31845–1

24. Schlegel M, Marquardt W. Detection and exploitation of the control switching structure in the solution of dynamic optimization problems. *Journal of Process Control* 2006; 16(3): 275-290. Selected Papers from Dycops 7 (2004), Cambridge, Massachusettsdoi: https://doi.org/10.1016/j.jprocont.2005.06.008

25. Wang P, Yang C, Yuan Z. The Combination of Adaptive Pseudospectral Method and Structure Detection Procedure for Solving Dynamic Optimization Problems with Discontinuous Control Profiles. *Industrial & Engineering Chemistry Research* 2014; 53: 7066-7078.

26. Chen W, Biegler LT. Nested direct transcription optimization for singular optimal control problems. *AIChE Journal* 2016; 62(10): 3611-3627. doi: https://doi.org/10.1002/aic.15272

27. Chen W, Ren Y, Zhang G, Biegler LT. A simultaneous approach for singular optimal control based on partial moving grid. *AIChE Journal* 2019; 65(6): e16584. doi: https://doi.org/10.1002/aic.16584

28. Stryk vO. *Numerical Solution of Optimal Control Problems by Direct Collocation*: 129–143; Basel: Birkhäuser Basel . 1993

29. Bulirsch R, Nerz E, Pesch HJ, Stryk vO. *Combining Direct and Indirect Methods in Optimal Control: Range Maximization of a Hang Glider*: 273–288; Basel: Birkhäuser Basel . 1993

30. Foroozandeh Z, Shamsi M, Pinho MDRD. A Hybrid Direct–Indirect Approach for Solving the Singular Optimal Control Problems of Finite and Infinite Order. *Iran J Sci Technol Trans Sci* 2018; 42: 1545–1554.

31. Chen Y, Huang J. A Continuation Method for Singular Optimal Control Synthesis. In: ; 1993: 1256-1260

32. Pager ER, Rao AV. Method for solving bang-bang and singular optimal control problems using adaptive Radau collocation,. Comput Optim Appl 81 (2022), 857–887. doi: https://doi.org/10.1007/s10589-022-00350-6

33. M. Alipour AHB. A hybrid parametrization approach for a class of nonlinear optimal control problems. *Numerical Algebra, Control Optimization* 2019; 9(4): 493-506.

34. Liao SJ. An approximate solution technique not depending on small parameters: A special example. *International Journal of Non-Linear Mechanics* 1995; 30(3): 371-380. doi: https://doi.org/10.1016/0020-7462(94)00054-E

35. SJ L. *Beyond perturbation: introduction to homotopy analysis method*. New York: Chapman and Hall/CRC. first ed. 2003

36. Jafarimoghaddam A. Surprising solutions for some challenging problems arising from boundary layer theory by a new technique: the homotopy contraction mapping technique (HCMT). *SN Appl. Sci.* 2019; 1(1104). doi: https://doi.org/10.1007/s42452-019-1114-z

37. Jafarimoghaddam A. On the Homotopy Analysis Method (HAM) and Homotopy Perturbation Method (HPM) for a nonlinearly stretching sheet flow of Eyring-Powell fluids. *Engineering Science and Technology, an International Journal* 2019; 22(2): 439-451. doi: https://doi.org/10.1016/j.jestch.2018.11.001

38. Bouremel Y. Explicit series solution for the Glauert-jet problem by means of the homotopy analysis method. *Communications in Nonlinear Science and Numerical Simulation* 2007; 12(5): 714-724. doi: https://doi.org/10.1016/j.cnsns.2005.07.001

39. Seywald H, Kumar RR. Method for automatic costate calculation. *Journal of Guidance, Control, and Dynamics* 1996; 19(6): 1252-1261. doi: 10.2514/3.21780

40. Enright PJ, Conway BA. Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. *Journal of Guidance, Control, and Dynamics* 1992; 15(4): 994-1002. doi: 10.2514/3.20934

41. Fahroo F, Ross IM. Costate Estimation by a Legendre Pseudospectral Method. *Journal of Guidance, Control, and Dynamics* 2001; 24(2): 270-277. doi: 10.2514/2.4709

42. Duan JS, Rach R. A new modification of the Adomian decomposition method for solving boundary value problems for higher order nonlinear differential equations. *Applied Mathematics and Computation* 2011; 218(8): 4090-4118. doi: https://doi.org/10.1016/j.amc.2011.09.037

43. Duan JS, Rach R, Wazwaz AM, Chaolu T, Wang Z. A new modified Adomian decomposition method and its multistage form for solving nonlinear boundary value problems with Robin boundary conditions. *Applied Mathematical Modelling* 2013; 37(20-21): 8687-8708. doi: 10.1016/j.apm.2013.02.002

44. Jafarimoghaddam A, Soler M, Simorgh A. The optimal decomposition method (ODM) for nonlinear problems. *Journal of Computational Science* 2022; 62: 101690. doi: https://doi.org/10.1016/j.jocs.2022.101690

45. Adomian G, Rach R. Nonlinear stochastic differential delay equations. *Journal of Mathematical Analysis and Applications* 1983; 91(1): 94-101. doi: https://doi.org/10.1016/0022-247X(83)90094-X

46. Saaty T, Bram J. *Non-linear Mathematics*. McGraw-Hill, New York . 1964.

47. Persson H. A fixed point theorem for monotone functions. *Applied Mathematics Letters* 2006; 19(11): 1207-1209. doi: https://doi.org/10.1016/j.aml.2006.01.008

48. ÉsikZoltán , RondogiannisPanos . A fixed point theorem for non-monotonic functions. *Theoretical Computer Science* 2015.

49. Glauert MB. The wall jet. *Journal of Fluid Mechanics* 1956; 1(6): 625–643. doi: 10.1017/S002211205600041X

50. Jafarimoghaddam A. Wall jet flows of Glauert type: Heat transfer characteristics and the thermal instabilities in analytic closed forms. *European Journal of Mechanics - B/Fluids* 2018; 71: 77-91. doi: https://doi.org/10.1016/j.euromechflu.2018.04.002

51. Ricciardi LA, Vasile M. Direct Transcription of Optimal Control Problems with Finite Elements on Bernstein Basis. *Journal of Guidance, Control, and Dynamics* 2019; 42(2): 229-243. doi: 10.2514/1.G003753

52. Miller AT, Hager WW, Rao AV. Mesh refinement method for solving optimal control problems with nonsmooth solutions using jump function approximations. *Optimal Control Applications and Methods* 2021; 42(4): 1119-1140. doi: https://doi.org/10.1002/oca.2719

53. Archibald R, Gelb A, Yoon J. Polynomial Fitting for Edge Detection in Irregularly Sampled Signals and Images. *SIAM Journal on Numerical Analysis* 2005; 43(1): 259-279. doi: 10.1137/S0036142903435259

54. Ganjefar S, Rezaei S. Modified homotopy perturbation method for optimal control problems using the Padé approximant. *Applied Mathematical Modelling* 2016; 40(15): 7062-7081. doi: https://doi.org/10.1016/j.apm.2016.02.039

55. Patterson MA, Rao AV. GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using Hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming. *ACM Trans. Math. Softw.* 2014; 41(1). doi: 10.1145/2558904

56. Aghaee M, Hager WW. The Switch Point Algorithm. *SIAM Journal on Control and Optimization* 2021; 59(4): 2570-2593. doi: 10.1137/21M1393315

57. Cots O, Gergaud J, Goubinat D. Time-optimal aircraft trajectories in climbing phase and singular perturbations. *IFAC-PapersOnLine* 2017; 50(1): 1625-1630. 20th IFAC World Congressdoi: https://doi.org/10.1016/j.ifacol.2017.08.327

9 | APPENDIX: PROOF OF THEOREM 3

Proof: The autonomous system of 1^{st} order differential equations over an arbitrary interval Ω_j is integrated with $\mathcal{L}^{-1}_{\Omega_j} = \int_{t_{j-1}}^{t} (\bullet) dt$ as,

$$\vec{x}_{\Omega_j}(t) = \vec{x}_{\Omega_j}(t_{j-1}) + \int_{t_{j-1}}^{t} \vec{f}\left(\vec{x}(t), \vec{c}_{\Omega_j}\right) dt \tag{124}$$

Following SADM, we assume,

$$\vec{x}_{\Omega_j}(t) = \sum_{i=0}^{\infty} \vec{x}_{i,\Omega_j}, \quad \vec{f}\left(\vec{x}(t), \vec{c}_{\Omega_j}\right) = \sum_{i=0}^{n\to\infty} \vec{\mathcal{A}}_i, \quad \vec{\mathcal{A}}_n = \frac{1}{n!}\frac{d^n}{dp^n}\left[\vec{f}\left(\vec{x}^{P}_{\Omega_j}; \vec{c}^{T}_{\Omega_j}\right)\right]_{p=0} \tag{125}$$

On using (125) into Eq. (124),

$$\begin{aligned} \vec{x}_{0,\Omega_j} &= \vec{x}_{\Omega_j}(t_{j-1}) \\ \vec{x}_{i,\Omega_j} &= \int_{t_{j-1}}^{t} \vec{\mathcal{A}}_{i-1} dt, \quad i \geq 1 \end{aligned} \tag{126}$$

Let us define an auxiliary vector $\vec{\mathcal{F}}_i\left(\vec{x}, \vec{c}_{\Omega_j}\right)$ as,

$$\begin{aligned} \vec{\mathcal{F}}_0\left(\vec{x}, \vec{c}_{\Omega_j}\right) &= \vec{f}\left(\vec{x}, \vec{c}_{\Omega_j}\right) \\ \vec{\mathcal{F}}_n\left(\vec{x}, \vec{c}_{\Omega_j}\right) &= J_a\left(\vec{\mathcal{F}}_{n-1}\left(\vec{x}, \vec{c}_{\Omega_j}\right)\right)\vec{\mathcal{F}}_0\left(\vec{x}, \vec{c}_{\Omega_j}\right), \quad n \geq 1 \end{aligned} \tag{127}$$

According to Eq. (126),

$$\begin{aligned} \vec{x}_{1,\Omega_j} &= \int_{t_{j-1}}^{t} \vec{\mathcal{A}}_0 dt = \int_{t_{j-1}}^{t} \vec{f}\left(\vec{x}, \vec{c}_{\Omega_j}\right)_{\vec{x}=\vec{x}(t_{j-1})} dt = \vec{f}\left(\vec{x}, \vec{c}_{\Omega_j}\right)_{\vec{x}=\vec{x}(t_{j-1})}\left(t - t_{j-1}\right) \\ &= \vec{\mathcal{F}}_0\left(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\right)\left(t - t_{j-1}\right) \end{aligned} \tag{128}$$

From Eq. (47), we have,

$$\vec{\mathcal{A}}_1 = J_a(\vec{f})\vec{f}\left(\vec{x}, \vec{c}_{\Omega_j}\right)_{\vec{x}=\vec{x}(t_{j-1})}\left(t - t_{j-1}\right) = \vec{\mathcal{F}}_1\left(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\right)\left(t - t_{j-1}\right) \tag{129}$$

Therefore,

$$\begin{aligned} \vec{x}_{2,\Omega_j} &= \int_{t_{j-1}}^{t} \vec{\mathcal{A}}_1 dt = \int_{t_{j-1}}^{t} J_a(\vec{f})\vec{f}\left(\vec{x}, \vec{c}_{\Omega_j}\right)_{\vec{x}=\vec{x}(t_{j-1})}\left(t - t_{j-1}\right) dt \\ &= \vec{\mathcal{F}}_1\left(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\right)\frac{\left(t - t_{j-1}\right)^2}{2!} \end{aligned} \tag{130}$$

From Eq. (50) and by making use of Eq. (128) and Eq. (130),

$$\vec{\mathcal{A}}_2 = \frac{\left(t - t_{j-1}\right)^2}{2!}\left[J_a(\vec{f})J_a(\vec{f}) + \sum_{l=1}^{L}\frac{\partial}{\partial x^{(l)}}J_a(\vec{f})f^{(l)}\left(\vec{x}, \vec{c}_{\Omega_j}\right)\right]\vec{f}\left(\vec{x}, \vec{c}_{\Omega_j}\right)\Big|_{\vec{x}=\vec{x}(t_{j-1})} \tag{131}$$

It can be shown that,

$$J_a(\vec{f})J_a(\vec{f}) + \sum_{l=1}^{L}\frac{\partial}{\partial x^{(l)}}J_a(\vec{f})f^{(l)}\left(\vec{x}, \vec{c}_{\Omega_j}\right) = J_a(J_a(\vec{f})\vec{f}) \tag{132}$$

Therefore,

$$\vec{\mathcal{A}}_2 = \vec{\mathcal{F}}_2\left(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\right)\frac{\left(t - t_{j-1}\right)^2}{2!} \tag{133}$$

and,

$$\vec{x}_{3,\Omega_j} = \int_{t_{j-1}}^{t} \vec{\mathcal{A}}_2 dt = \vec{\mathcal{F}}_2\left(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\right)\frac{\left(t - t_{j-1}\right)^3}{3!} \tag{134}$$

Following the same procedure for the perturbed vectors of higher orders, we obtain,

$$\begin{aligned}
\vec{x}_{4,\Omega_j} &= \int_{t_{j-1}}^{t} \vec{\mathcal{A}}_3 dt = \vec{\mathcal{F}}_3\big(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\big)\frac{\big(t-t_{j-1}\big)^4}{4!} \\
&... \\
\vec{x}_{i,\Omega_j} &= \int_{t_{j-1}}^{t} \vec{\mathcal{A}}_{i-1} dt = \vec{\mathcal{F}}_{i-1}\big(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\big)\frac{\big(t-t_{j-1}\big)^i}{i!}
\end{aligned} \tag{135}$$

Since, $\vec{x}_{\Omega_j} = \vec{x}_{0,\Omega_j} + \vec{x}_{1,\Omega_j} + ... + \vec{x}_{n,\Omega_j}$,

$$\vec{x}_{\Omega_j}(t) = \vec{x}(t_{j-1}) + \vec{\mathcal{F}}_0\big(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\big)\big(t-t_{j-1}\big) + ... + \vec{\mathcal{F}}_n\big(\vec{x}(t_{j-1}), \vec{c}_{\Omega_j}\big)\frac{\big(t-t_{j-1}\big)^{n+1}}{\big(n+1\big)!}\blacksquare \tag{136}$$

Corollary 4. for an autonomous system of 1^{st} order differential equations as defined in theorem, the global recursion scheme for Adomain vectors is,

$$\begin{aligned}
\vec{\mathcal{A}}_0 &= \vec{f}\big(\vec{x}_{0,\Omega_j}, \vec{c}_{\Omega_j}\big) \\
\vec{\mathcal{A}}_n &= \frac{\big(t-t_{j-1}\big)}{n} J_a\big(\vec{\mathcal{A}}_{n-1}\big)\vec{\mathcal{A}}_0
\end{aligned} \tag{137}$$

Proof: From Eq. (135), we can write,

$$\vec{\mathcal{A}}_i = \vec{\mathcal{F}}_i\big(\vec{x}_{0,\Omega_j}, \vec{c}_{\Omega_j}\big)\frac{\big(t-t_{j-1}\big)^i}{i!} \tag{138}$$

Therefore, we have,

$$\vec{\mathcal{A}}_{n-1} = \vec{\mathcal{F}}_{n-1}\big(\vec{x}_{0,\Omega_j}, \vec{c}_{\Omega_j}\big)\frac{\big(t-t_{j-1}\big)^{n-1}}{n-1!} \tag{139}$$

and,

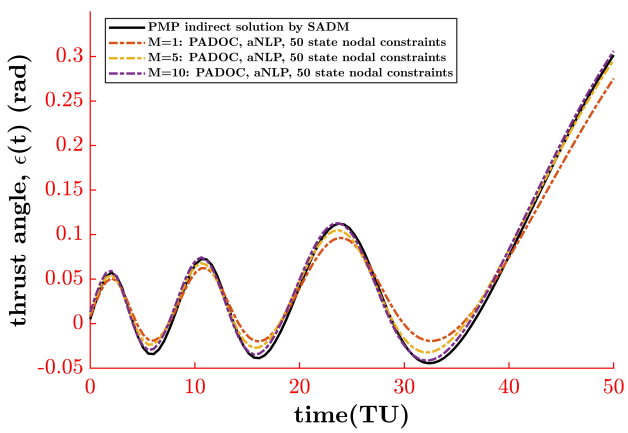
$$\vec{\mathcal{A}}_n = \vec{\mathcal{F}}_n\big(\vec{x}_{0,\Omega_j}, \vec{c}_{\Omega_j}\big)\frac{\big(t-t_{j-1}\big)^n}{n!} = J_a(\vec{\mathcal{F}}_{n-1})\vec{\mathcal{A}}_0\frac{\big(t-t_{j-1}\big)^n}{n!} \tag{140}$$

From Eq. (139) and Eq. (140),

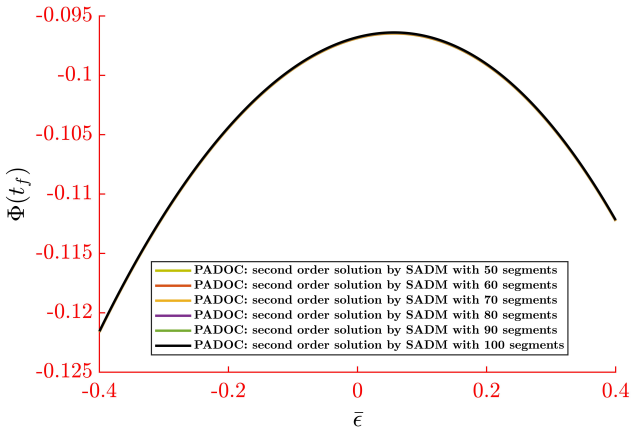
$$\vec{\mathcal{A}}_n = \frac{\big(t-t_{j-1}\big)}{n} J_a\big(\vec{\mathcal{A}}_{n-1}\big)\vec{\mathcal{A}}_0\blacksquare \tag{141}$$

Remark 2. The Global Analytic solution presented in theorem 3 can be extended to a large class of non-autonomous n^{th} order nonlinear ODEs. However, derivation of such a formula is not within the scope of the present work.

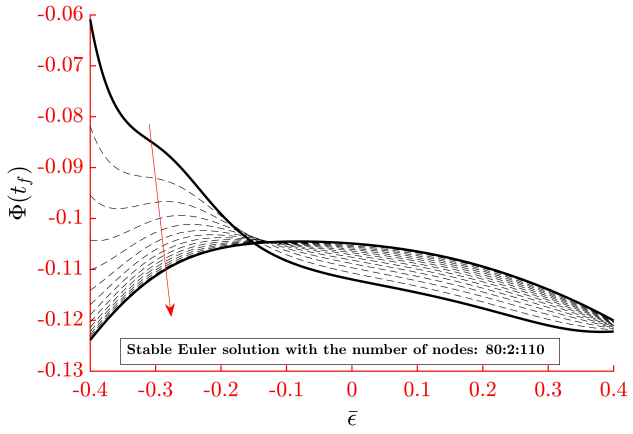
□



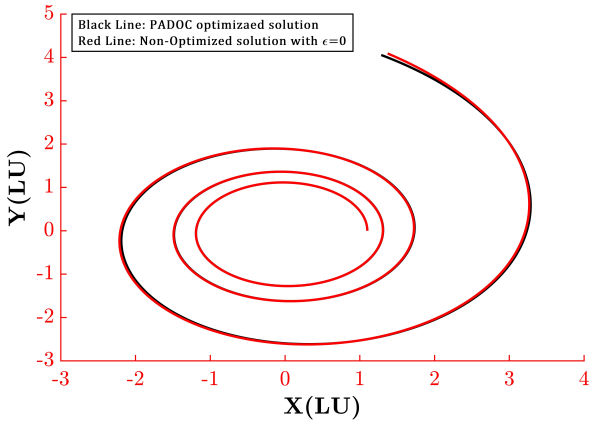
(a)



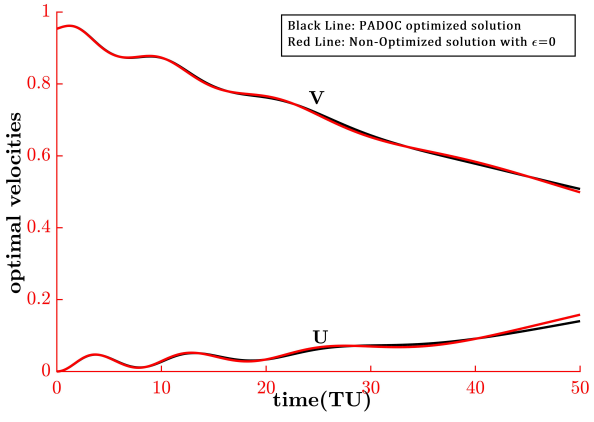
(b)



(c)

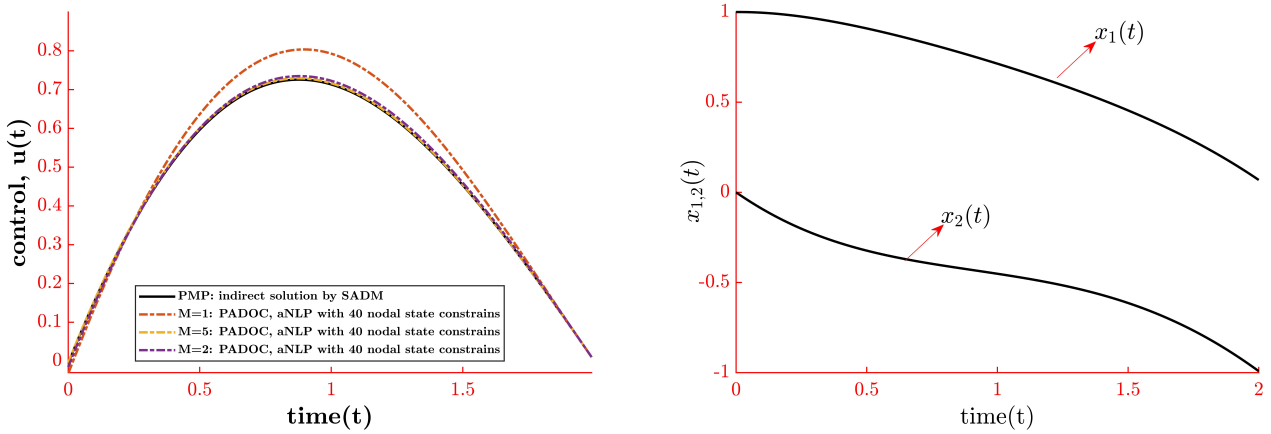


(d)



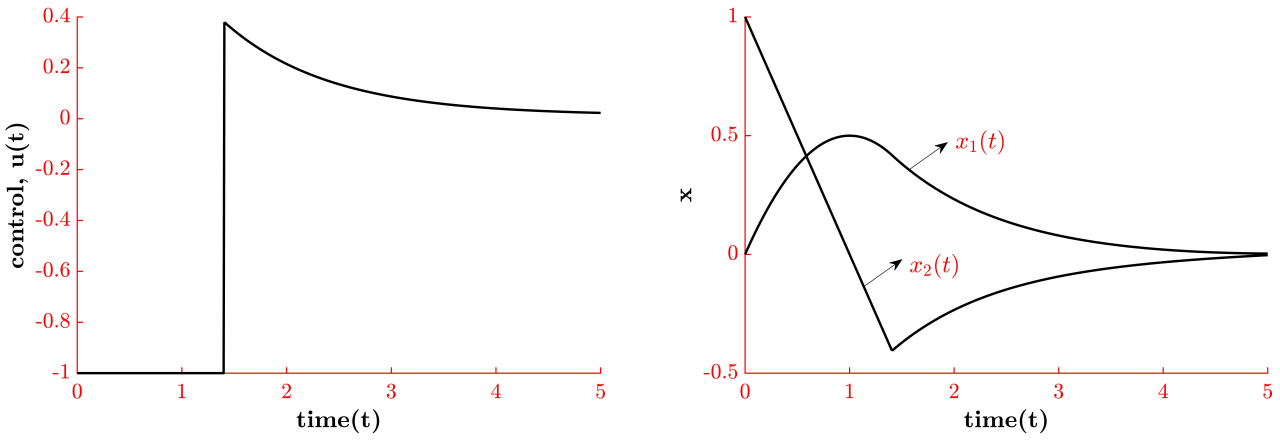
(e)

FIGURE 1 Example 7.1.(a): the converted aNLP solutions compared to the PMP solution. (b): the objective in PADOC aNLP with $M = 1$ as a function of control. (c):the objective in Euler-based transcription with $M = 1$ as a function of control. (d) and (e): state variables from $\epsilon = 0$ (red lines), compared to those from PADOC algorithm after optimization (black lines)



(a) (b)

FIGURE 2 Example 7.2. (a): the converted aNLP solutions compared to PMP solution. (b): the state variables from PADOC complete solution



(a) (b)

FIGURE 3 Example 7.4. (a): the optimal control from PADOC algorithm. (b): the optimal state variables from PADOC algorithm

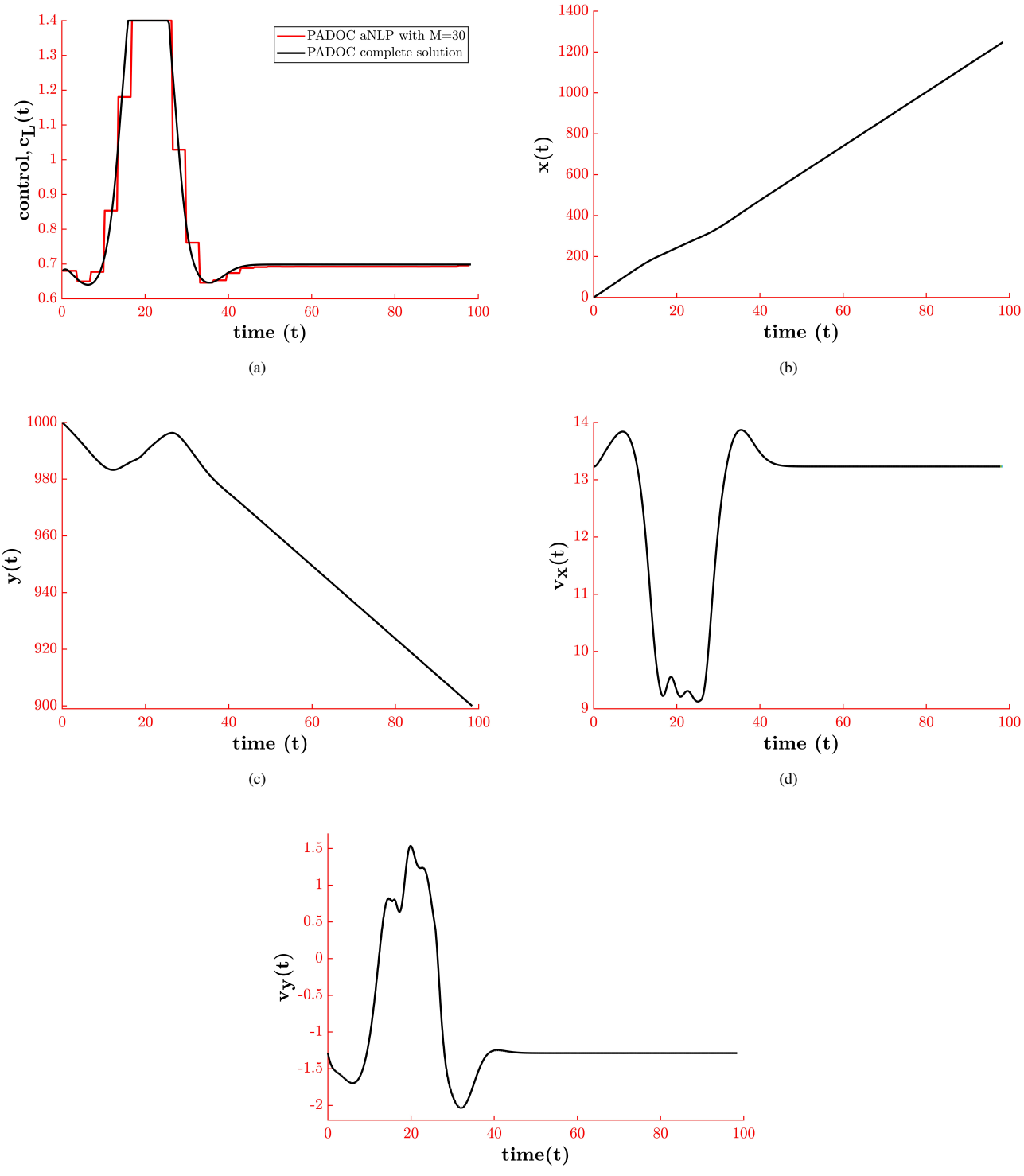


FIGURE 4 Example 7.5. (a): the PADOC aNLP with $M = 30$ (red line) compared to the PADOC complete solution (black line). (b),(c),(d),(e): optimal state variables from the PADOC complete solution

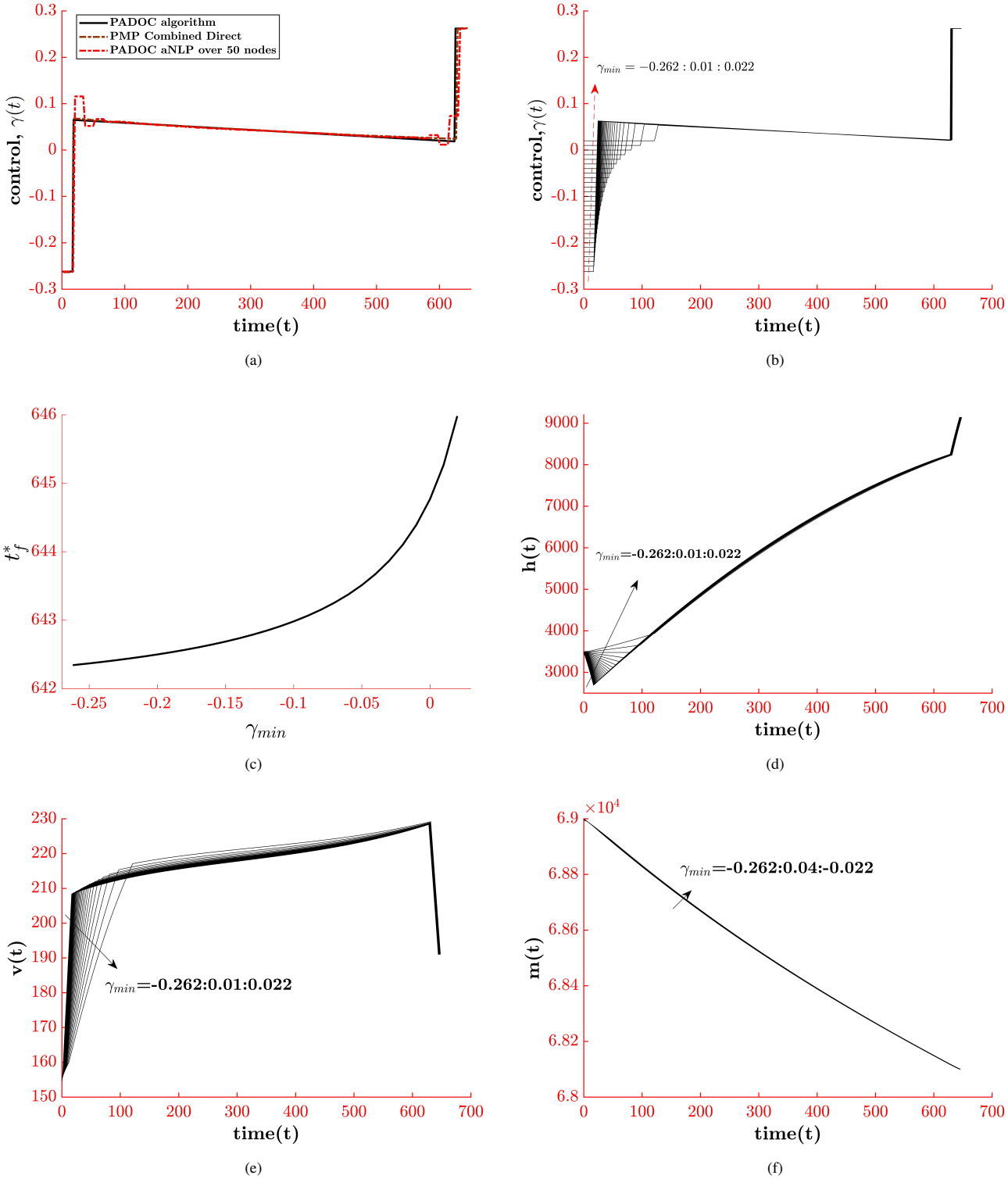


FIGURE 5 Example 7.6. (a): PADOC aNLP with $M = 50$ compared to the PADOC complete solution and PMP combined direct method. (b): the PADOC complete solutions in varying γ_{min}. (c): minimum time as a function of γ_{min}. (d),(e),(f): the optimal state variables in varying γ_{min}